



## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :

G06F 13/00

A1

(11) International Publication Number:

WO 99/48011

(43) International Publication Date: 23 September 1999 (23.09.99)

(21) International Application Number: PCT/US99/06074

(22) International Filing Date: 19 March 1999 (19.03.99)

(30) Priority Data:

09/045,280

20 March 1998 (20.03.98)

US

(71) Applicant: ADVANCED WEB SOLUTIONS, INC. [US/US];  
Suite 110, 2171 Junipero Serra Boulevard, Daly City, CA  
94014 (US).(72) Inventor: MAURILLE, Mithras, C.; 57 North Street, San  
Francisco, CA 94112 (US).(74) Agents: TEST, Aldo, J. et al.; Flehr, Hohbach, Test, Albritton  
& Herbert LLP, Suite 3400, 4 Embarcadero Center, San  
Francisco, CA 94111-4187 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

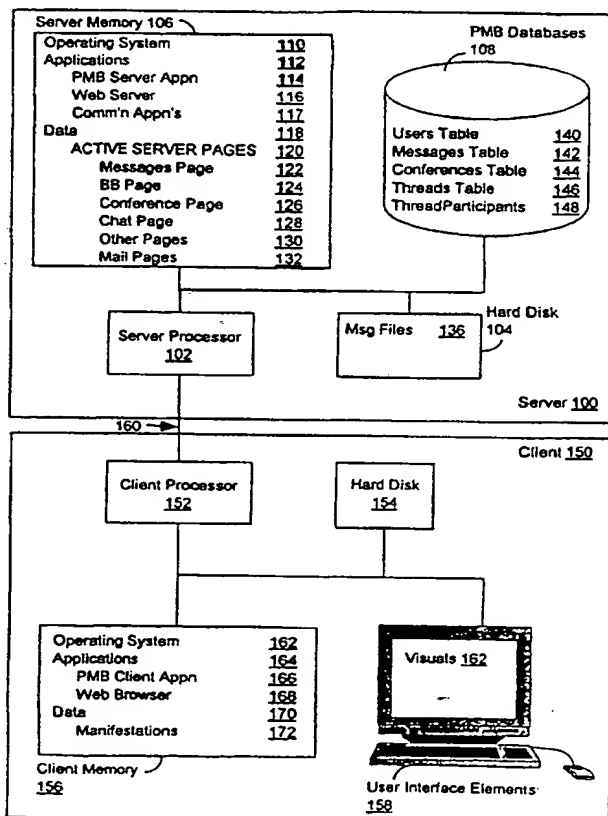
Published

With international search report.

(54) Title: COMMUNICATION BOARD SYSTEM AND METHOD FOR USE IN COMPUTER NETWORKS

## (57) Abstract

A system and method that provides integrated combinations of threaded instant messages, open display bulletin boards, private bulletin boards, threaded e-mail, explicit acknowledgment of messages, and conferencing modes. The system can be implemented in any Internet-based computer network (160), including the Internet, intranets, and extranets. System components include a server application (100), a client application (150) and a data repository (108) maintained by the server (100). The server records in the data repository pertinent information regarding communications between and requests issued by users (158), and handles the communications cooperatively with the client (150) in accordance with the system mode being exercised (e.g., talk, conferencing, whisper, mail, messaging, open display, private bulletin boards, etc.).



*FOR THE PURPOSES OF INFORMATION ONLY*

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon		Republic of Korea	PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakhstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

## COMMUNICATION BOARD SYSTEM AND METHOD FOR USE IN COMPUTER NETWORKS

The present invention relates generally to electronic communication systems for use in computer networks and, particularly, to bulletin boards, instant messaging systems, electronic mail and chat rooms.

5

### BACKGROUND OF THE INVENTION

An electronic communication system for use in enterprise and other collaborative environments would ideally include a suite of capabilities that facilitate decision making and communication by two or more individuals. Such capabilities could include:

- enabling users to view the history of multiple conversations with multiple parties (referred to hereinafter as "conversation history");
- enabling users to view messages as soon as they are available without requiring the users to log onto a public bulletin board system (BBS) (referred to hereinafter as "instant access");
- enabling users to view the content of messages without requiring that the messages first be selected (referred to hereinafter as "open display");
- enabling users to conduct their conversations in privacy so that each user is the only person who can view the history and content of their respective multiple conversations (referred to hereinafter as "private conversations");

- enabling users to undeniably agree to proposals made in the course of a conversation in such a way that the conversation is concluded (referred to hereinafter as "agreement"); and
- enabling users to participate in moderated conferences or informal chats, as well as in conversations (referred to hereinafter as "integrated modes").

Prior art electronic systems, which include electronic mail (e-mail), bulletin board systems (BBS), instant messaging and chat rooms, offer some but not all of these capabilities and, as a result, are less than ideally suited to enterprise communications. The capabilities of these various communication systems are presented in Table 1.

TABLE 1

System	Conversation History	Instant Access	Open Display	Private Conversations	Agree't	Integrated Modes
E-mail	No	Yes	No	Yes	No	No
BBS	Yes	No	No	No	No	No
Instant Messaging	No	Yes	Yes	Yes	No	No
Chat	No	Yes	Yes	No	No	No

Referring to Table 1, e-mail systems offer instant access to messages, but the messages must be selected and opened by the recipient (typically with a mouse click) to be viewed. E-mail messages are private as they are directed to specific recipients (one or many). No viewable history is available for E-mail messages except for the most rudimentary kind wherein a message being replied to can be copied into the body of the response. E-mail has only one mode and includes no feature whereby an e-mail user can issue a formal agreement to a proposal contained in a message, other than so stating in a reply message; e.g., "I agree to your proposal of 12/10/97 on the subject of contract terms."

Like e-mail systems, bulletin board systems (BBS) do not provide open display, agreement, or integrated mode capabilities. Unlike E-mail systems, BBS display messages in a threaded format wherein a topic is listed first and all messages germane to that topic are listed below the topic with different levels of indentation indicating historical and logical relationships between the related messages. For example, a BBS might display a topic and two messages on the topic, the first having an associated comment, as follows:

**Topic: Discussion of X**

**Message 1**

**Comment on Message 1**

**Message 2**

Because BBS do not provide open display, a user must select a message or comment to read that message or comment. BBS are centralized systems, meaning that a user must actually log in to the BBS to view topics and messages. As a result, messages are not immediately accessible (i.e., to read messages on a topic of interest a user must first access the BBS). Also, BBS are anything but private; typically, the same bulletin board is available to all users, meaning that all messages are accessible to all users.

Instant messaging systems allow users to communicate privately in real time over a network connection. An example of such a system is America On Line's "Instant Messages" feature, which allows an AOL member who is online to communicate with another member who is also online at the same time. Messages are openly displayed (i.e., without needing to be selected first). Thus, instant messaging systems provide private conversations, immediate access and open display capabilities. However, instant messaging systems do not support conversation history, agreement or integrated modes. Moreover, instant messaging systems are for one-on-one communication only.

Chat systems allow a group of users to enter a chat room and then engage in a group conversation. The group conversation can be moderated or un-moderated. Like instant messaging, chat systems are for informal communications and do not provide conversation history, agreement or integrated modes. Also like instant

messaging, chat systems openly display all messages. However, chat messages are not immediately accessible as a user needs to enter a chat room before they can view messages. Also, chat rooms are not private as anyone in the chat room can read all chat messages typed by users in that room.

5

Consequently, there is a need for an enterprise communication system that provides features and/or combinations of features not present in prior art electronic communication systems.

10

### SUMMARY OF THE INVENTION

In summary, the present invention is a electronic communication system that provides features and/or combinations of features not present in prior art electronic communication systems. In particular, the present invention is a communication board system with multiple modes in which the communication board system can be variously configured as:

15

20

25

- a threaded instant message system (conversation history plus instant access capabilities);
- an open display bulletin board system (conversation history plus open display capabilities);
- private message boards (conversation history plus private conversations capabilities);
- a system allowing message locking (conversation history plus agreement capabilities); and
- a threaded mail system.

30

The preferred embodiment of the system is implemented as a client server system including a server application, a client application and a data repository resident on the server. In the preferred embodiment a sender requests communication services using his client application, which relays the request to the server program. The server program updates the data repository to reflect the request and then issues an appropriate message to the client application of one or more recipients, depending

on the requested communication services. Any response by a recipient is cooperatively handled by the client and server applications in the same manner as the initial request. All communications activities are moderated by the server application and recorded in the data repository.

5

The data repository is preferably structured as a set of relational database tables, including a users table and a messages table. The users table lists characteristics of all system users, including unique ID, name and preferences. The messages table lists characteristics of each message issued by users, including a unique message ID, threading information (parent and child message IDs) sender and recipient name, subject, status of the message (unread, read, responded, acknowledged, etc. ), type of message (thread, invitation, confirmation, log), miscellaneous flags, and the name of the file where the message text is stored on the server.

15

In the preferred embodiment a sender sends a message to a recipient by filling in a send message template displayed by the client application, which relays the completed message information to the server application. Upon receiving the message information the server application stores the pertinent information (sender, recipient, subject) in the message table, updates message status fields and threading information for the same message table record, stores the message text in a file, and issues the client application of the intended recipient a pending-message alert. Preferably, the server application sends the pending-message alert only when the users table indicates that the recipient is online, although this is not a requirement of the present invention. The client application gives the recipient an opportunity to do nothing, cancel the alert, or accept the message. If the user does nothing, the server resends the alert at some prescribed interval. If the user cancels the alert, the server will not resend the alert and places the message on hold. If the user accepts the message, the server sends the relevant message information to the client application to be accessed by the recipient.

20

25

30

When the preferred embodiment is configured as a private message board system, each user interacts with the communication system via a private bulletin board in

which the client application instantly displays the history and content of all messages associated with conversations in which the respective user is a party. In this configuration the present invention supports multiple instances of instant messaging, meaning that it provides private bulletin boards for multiple users and is able to allow each of the multiple users to exchange instant messages with one or more other users. In this case the relevant message information returned by the server application to the intended recipient includes threading information, which enables the client application to display the message's history along with that of other messages. Preferably, the client application displays the messages in open format; however, the client application could also display the messages in the conventional format.

Once a recipient has accepted a message, he can reply to or acknowledge the message. The message reply feature is implemented cooperatively by the client and server applications in the same manner as the message send feature. Message acknowledge is a feature of the present invention wherein a conversation's thread is closed, indicating that the conversation has been concluded, typically with some agreement. For example, a user can indicate final acceptance of sales terms set out in the last of several threaded messages by acknowledging that message. The client application relays message acknowledge information to the server application, which updates the data repository by setting the message status to *acked* and closing the corresponding thread. Thus, the present invention allows the entire history of a negotiation to be preserved along with its final agreement.

The other system configurations can be implemented using the components of the preferred embodiment with minimal changes from the above description.

In the preferred embodiment, the client application is based on a web browser and the server application includes communication board software that performs all high level and data repository operations and web server software that decodes and encodes communications from and to the client web browser. Preferably, all communication board contents displayed to users are formatted as ACTIVE



SERVER PAGES whose fields can be dynamically filled in by the user via the client web browser or by the server's communication board software using information from other users and/or the data repository.

- 5 Another feature provided by the preferred embodiment is message hyperlinking, which allows a user to form a response to any message shown on their private bulletin board by simply selecting a hypertext link identifying the message sender. In response to the selection of a hyperlink the client application generates the appropriate response screen with some of the fields (e.g., sender, recipient, subject)  
10 filled-in.

- The preferred embodiment supports multiple integrated modes, including a threaded communications (message) mode, already described, talk mode, conference mode, whisper mode and mail mode. The present invention allows users to transition  
15 smoothly between the modes.

- Conference mode allows users to participate in moderated or unmoderated conferences that are scheduled or unscheduled. Information regarding conference participants and scheduled time and moderator, if applicable, are stored by the  
20 server application in a conferences table within the data repository. When a user is to participate in a scheduled conference he is notified of the virtual conference room and time of the conference by the server application. The server application also creates the virtual conference room at the appropriate time, registers participants, and stores a log of the conference in the data repository. In a preferred  
25 embodiment conference mode conversations are unthreaded. Whisper mode is available to participants in a conference who wish to participate in a private, side conversation.

- Talk mode allows users to participate in informal, unlogged conversations. In a  
30 preferred embodiment talk mode conversations are unthreaded.

Mail mode allows a user to send e-mail over the Internet with two levels of threading.

## BRIEF DESCRIPTION OF THE DRAWINGS

Additional objects and features of the invention will be more readily apparent from  
5 the following detailed description and appended claims when taken in conjunction  
with the drawings, in which:

FIG. 1 is a block diagram of a preferred embodiment of the present invention;

10 FIG. 2 is a block diagram of the database tables 140, 142, 144, 146, 148 from FIG.  
1;

FIG. 3A is a data flow diagram illustrating the relationships between a client  
application 166 and web browser 168, the web server 116, the server application  
15 114 and the PMB databases 108 when a user is sending a message;

FIG. 3B is a block diagram of new records allocated in the various tables by the  
server application 114 while processing an exemplary send request;

20 FIG. 4A is a data flow diagram illustrating steps performed by the web browser 168-  
1 of a message sender, the server application 114 and the web browser 168-2 of a  
message recipient for a message send procedure;

FIG. 4B is an image of a private communication board screen on which instant  
25 messages owned by one user are presented in a threaded, open format;

FIG. 4C is an image of a second private communication board screen on which  
instant messages owned by a second user are presented in a threaded, open  
format;

30 FIG. 4D is an image of a private message review board that presents for one user  
all of the user's messages having a common subject;

FIG. 5A is a data flow diagram illustrating steps performed by a web browser 168-1 of a message replier, the server application 114, and a web browser 168-2 of a message reply recipient for a message reply procedure;

- 5 FIG. 5B is a block diagram of new records allocated in the various tables by the server application 114 while processing an exemplary reply request;

- 10 FIG. 6 is a data flow diagram illustrating steps performed by the web browser 168-1 of a message acknowledger and the server application 114 for a message acknowledge procedure;

FIG. 7A is a flow chart illustrating steps by which the server application 114 schedules and implements a conference;

- 15 FIG. 7B depicts an exemplary communications board for a particular user illustrating conference notifications and announcements;

FIG. 7C depicts a conference screen displayed by client Web browsers 168 for all clients participating in a conference;

20

FIG. 8A is depicts a talk entry screen displayed by client Web browsers enabling a user to transition to talk mode;

- 25 FIG. 8B is depicts a talk session screen displayed by client Web browsers during a talk session; and

FIG. 9 depicts a mail screen displayed by client Web browsers that supports threaded Internet e-mail.

30

## DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring to FIG. 1, there is shown a block diagram of a preferred embodiment of the present invention. This embodiment includes a server 100 with a server memory 106, processor 102, hard disk 136 and database 108. The server memory 106 could be any combination of a RAM or cache memory and includes an operating system 110, application programs 112 and data 118. In accordance with well known principles, the processor 102 executes the applications 112 in the memory 106 under control of the operating system 110.

The applications 112 include a personal message board (PMB) server application embodying many of the teachings of the present invention and a web server 116, which could be any program configured to serve web content over a network. The applications 112 also include communications application 117, which are programs that employ features of the server application 114. The data 118 include ACTIVE SERVER PAGES (ASP) 120 that describe the contents of web pages through which clients 150 exercise the modes of the present invention, including messaging, conferencing (incorporating a whisper mode for conference participants), talk, and mail. The ASP 120 therefore includes a messages page 122, a conference page 124, a whisper page 126, a chat page 128, a mail page 130 and other pages 132 needed for miscellaneous client-server communications.

**Message mode** allows a user to interact with a private bulletin board in which his messages (i.e., any message involving the user as sender or recipient) are instantly available and displayed with full threading information. Message mode supports an acknowledge (Ack) reply which, when sent by a user in response to a particular message, closes the thread comprising the particular message and records the users acknowledgment that they read/accepted the message. Because each bulletin board is private, no user other than the authorized user can view its contents.

**Conference mode** allows users to participate in moderated or unmoderated conferences that are scheduled or unscheduled. Information regarding conference

participants and scheduled time and moderator, if applicable, are stored by the server application in a conferences table within the data repository. When a user is to participate in a scheduled conference he is notified of the virtual conference room and time of the conference by the server application. The server application also  
5 creates the virtual conference room at the appropriate time, registers participants, and stores a log of the conference in the data repository. In a preferred embodiment conference mode conversations are unthreaded. Users involved in a conference can enter **whisper mode**, in which they can converse privately, without logging.

10 **Talk mode** allows users to participate in informal, unlogged conversations. In a preferred embodiment talk mode conversations are unthreaded. **Mail mode** allows a user to send threaded e-mail over the Internet. Each of these modes is integrated so that users can transition from one to the other.

15 In the preferred embodiment, the database 108 is a relational database system including tables organized to store information written and retrieved by the server application 114 in the course of its operation. The database tables include a users table 140, messages table 142, conference table 144, threads table 146 and a  
20 thread participants table 148. The tables 140-148 respectively store information on: system users (140), messages exchanged between users (142), conferences of users (144), mapping of messages to threads (146), and mapping of thread participants (users) to threads (148). These tables are described in depth in reference to FIG. 2. The hard disk 104 includes message files 136, which store the  
25 text of messages referred to by the messages table.

The embodiment shown in FIG. 1 also includes one or more clients 150, each of which is used by one or more users. The single client 150 shown is representative of the one or more possible clients. In the preferred embodiment, clients 150 are  
30 coupled to the server 102 via an intranet 160; however, the principles of the present invention are equally applicable to any type of network, such as the Internet, an extranet or combinations thereof. The client 150 and the server 100 exchange information over the network 160 using standard network protocols, such as TCP/IP

and HTTP. In particular, the server application 114 makes available to users communication board services (encompassing private message boards, chat, talk, conferencing and mail) by transmitting to the clients 150 via the web server software 116 manifestations of the ACTIVE SERVER PAGES 120 and other web pages.

- 5 These manifestations, when displayed by the clients 150, enable the users to view communications board information and messages from other users, and to issue requests and queries to the PMB application 114 via the web server software 116.

- 10 The client 150 includes a memory 156, processor 152, hard disk 154 and user interface elements 158, such as a display, keyboard and selection device. The memory 156 could be any combination of a RAM or cache memory and includes an operating system 162, application programs 164 and data 170. In accordance with well known principles, the processor 152 executes the applications 164 in the memory 156 under control of the operating system 162.

15

- The applications 164 include a personal message board (PMB) client application 166 and a web browser 168. The web browser 168 receives, transmits and presents manifestations from the ASP 120 and other pages transmitted over the web by the server application 114 via the web server 116. The client application 166
- 20 provides additional processing capabilities not present in the web browser 168 to assist users in interacting with the communication board features. These additional processing capabilities can include: responding to alerts from the PMB application 114 when the user is not available, locally logging user sessions, maintaining a local address book for the user and issuing standard queries or requests to the server
- 25 application 114. Note that simple implementations of the present invention can eliminate the PWB server application 166; in such an implementation all user interaction would be provided by the web browser 168. Because communications between the server 100 and clients 150 adhere to web standards and the key components of the present invention (i.e., the PMB server application 114 and PMB
- 30 client applications 166, described below) are compatible with standard web software (i.e., the web server 116 and browser 168), the present invention can be implemented in any web based client server environment. Moreover, with slight

modification, the server and client applications 114, 166 could be made compatible with any standard client/server communications packages (e.g., Novell, etc.).

5 The data include manifestations 172 of the ACTIVE SERVER PAGES (ASP) 120 downloaded by the web browser 168 in response to events or user requests. These manifestations (not shown in detail) enable the user to employ and interact with the features of the PMB server application, including the following modes: messaging, bulletin board, conferencing, chat, and mail. The manifestations are presented by the web browser 166 to users as visuals 162 and/or sounds 164 using the user  
10 interface elements 158.

The present invention is not limited to the described embodiment. In fact, the teachings of the present invention are applicable to any combination of current and/or future technology similar to that described herein. For just one example, the  
15 database 108 can be implemented using an object-oriented DBMS, flat text files stored on a hard disk 104, or other DBMS or non-DBMS solutions. Having generally described the preferred embodiment, the tables managed by the database system 108 are now described in reference to FIG. 2.

20 Referring to FIG. 2, there is shown a block diagram of the database tables 140, 142, 144, 146, 148 from FIG. 1. The users table 140 holds information pertaining to authorized users of the communication board (i.e., the server application 114). The **users table 140 fields** include:

25	<b>Id</b>	Unique number automatically assigned to each user. [primary key]
	<b>Name</b>	Assigned name for user's account. Used to logon with.
	<b>DisplayName</b>	User's name as it appears in the heading of their Comm Board.
	<b>Password</b>	To restrict account access by unauthorized users.
30	<b>SortPrefs</b>	User's sorting preferences - a 3 character code: c=chronological, r=reverse chronological, s=subject, e=sender

<b>ExpiryPref</b>	Number of days after which a message expires and is deleted.
<b>HasNewMail</b>	Flag is set to true when user has unread mail.
<b>HasInvitation</b>	Flag is set to true when user has been invited to join a conference.
<b>HoldAlerts</b>	Flag is true if user does not want alerts for new messages, etc to interrupt their session.

The messages table 142 holds information pertaining to individual messages. Each participant in a message owns an individual message record. This allows for personalized manipulation of a given message. The messages table 142 fields include:

<b>MsgId</b>	Unique number assigned automatically to each message. [primary key]
<b>MsgFileName</b>	Name of file that contains message body text.
<b>ThreadLevel</b>	Number 1-6 designating message's level in thread.
<b>ThreadId</b>	Id of corresponding Threads table record.
<b>ParentId</b>	Id of message that is above this message in the message thread. Value is zero if message is at top of thread.
<b>ChildId</b>	Id of message that is below this message in the message thread. Value is zero if message is at the bottom of thread.
<b>MsgTimeStamp</b>	Year, month, day, hour, minute and second when message was sent.
<b>MsgRecOwner</b>	Name of user that owns this message record (sender or recipient).
<b>SenderId</b>	User Id of message sender. Corresponds to id in Users table.
<b>SenderName</b>	User name of message sender.
<b>SenderNameASCII</b>	SenderName sort number.
<b>Recipient</b>	User name of message recipient.
<b>RecipientList</b>	Complete comma delimited list of recipients.
<b>CcList</b>	Complete comma delimited list of carbon copy recipients.



	<b>IsCarbonCopy</b>	Flag is true if recipient is in ccList rather than recipientList.
	<b>Subject</b>	Message subject or conference name.
	<b>SubjectASCI</b>	Subject sort number.
5	<b>AckDate</b>	Date message was acknowledged rather than responded to.
	<b>IsRespondedTo</b>	Flag is true if message has been responded to.
	<b>Status</b>	Message status contains one of these values: unread, read, repliedTo, acked, stored, deleted, abandoned.
10	<b>Type</b>	Type of message contains one of these values: thread, announcement, invitation, confirmation or log. A thread type is a message that appears as part of a thread. An announcement is a simple message sent to all users by default. Announcements do not allow for a reply. An invitation is automatically sent to users that are invited to a conference. A confirmation is automatically generated when a user responds to an invitation. A log is a complete record of a given conference.
15		
	<b>ExpiryDate</b>	Date message is to be expired (automatically deleted).
20	<b>IsForward</b>	Flag is true if message is forwarded.
	<b>ForwardThreadId</b>	Id of thread being forwarded.

A thread includes a first level message and subsequent replies, which can be added to the thread until the thread is closed by one of the thread participants issuing an explicit acknowledgment to one of the thread's messages. The threads table contains one thread record for each message thread that is unique by subject. The threads table 146 fields include:

30	<b>ThreadId</b>	Unique number assigned automatically to each thread. [primary key]
	<b>Subject</b>	Thread subject.
	<b>SubjectASCI</b>	Sort number for subject.

Whenever a user participates in a thread - that is, when he is a sender or recipient (even if only a CC recipient) - an entry is made by the server application 114 in the thread participants table 148. This table facilitates the gathering of message threads by subject for a given user. For example, the server application 114 would

5 gather such information for a particular user by:

- 1) issuing a query in the ThreadParticipants table 148 to find ThreadIds of all threads a user with a particular UserID has participated in;
- 2) issuing a query in the Threads table 146 using the ThreadIDs from query 1 to identify the subject sort numbers (SubjectASCI) for the
- 10 respective threads; and
- 3) issuing a query in the Messages table 142 with the SubjectASCI values from query 2) to identify all messages with the same subject for the user.

15 The thread participants table 148 fields include:

<b>ParticipantId</b>	Unique number assigned automatically to each thread. [primary key]
<b>ThreadId</b>	Thread ID (correlated with thread subject).
<b>UserId</b>	ID of user who is thread participant.

20 The conferences table 146 holds basic information about each conference. The conferences table 144 fields include:

<b>Id</b>	Unique number assigned automatically to each conference. [primary key]
25 <b>ModeratorId</b>	User id of conference moderator.
<b>Topic</b>	Moderator assigned topic for conference.
<b>Type</b>	Type of conference contains one of these values: private, public.
<b>Participants</b>	Comma delimited list of users invited to join a private
30 <b>IsScheduled</b>	Flag is true if the conference is scheduled for a later date as opposed [to?] immediately.

**When** Time of conference if the conference is scheduled for a later time.

Referring to FIG. 3A, there is shown a data flow diagram illustrating the relationships between a client application 166 and web browser 168, the web server 116, the server application 114 and the PMB databases 108 when a user is sending a message. The progression of operations shown in FIG. 3 is preceded by a SEND request issued by a user that is relayed via the user's web browser 168 to the sender application 114 via the web server 116. The sender application 114 in response returns a manifestation of a SEND page, which the web browser 168 displays as a SEND page visual 162. The user then completes at least a subset of the fields of the SEND page, which include: TO (i.e., one or more recipients), FROM (i.e., sender name), SUBJECT (i.e., message subject) and MSG (i.e., message text). It is assumed for this example that the message being sent is a level one message that starts a new thread of conversation.

The progression shown in FIG 3A. begins when the user sends the message by selecting the displayed SEND button on the visual (3.1). In response to the selection of the SEND button (3.1) the web browser 168 (possibly with some intervention of the client application 166) issues an HTTP message transferring the SEND data to the web server 116. In accordance with the HTTP (hypertext transfer protocol) the browser 168 transmits the SEND data to the web server 116 (3.2). The message (3.2) includes the URL of the page for which the data is being transmitted (i.e., "SendMsg") and the encoded data. Upon receiving the message (3.2) the web server 116 decodes, re-packages, and transmits (3.3) the data to the server application 114.

Upon receiving the message (3.3), the server application 114 performs the following operations (3.4-3.14):

- 3.4) Determines from the users table 140 the IDs of sender and recipient(s).
- 3.5) Writes the MsgTxt to the hard disk (3.5) as a message file 136 and generates a corresponding file name (MsgFileName).

3.6) Generates a unique ID (MsgID) and sortable subject number (SubjectASCI) for the message.

3.7) Allocates in the messages table 142 2N message records 143, N for the sender 143s and 1 for each of the N recipients 143ri (where i is an integer between 1 and N);

3.8) Updates each message record in accordance with the message data (only selected field are shown):

MsgID = message ID generated by server app. 114;

MsgFileName = message name generated by server app. 114;

ThreadLevel = 1 (message is at the top of a thread);

ParentID = 0 (message is at the top of a thread);

ChildID = 0 (message is also at the bottom of a thread);

MsgRecOwner = name of a respective one of the participants

SenderID = ID of sender;

SenderName = user name of sender;

Recipient = user name of a respective one of the N recipients;

RecipientList = list of N recipients;

Subject = subject completed by sender;

SubjectASCI = subject sort number generated by server app.

AckDate = 0 (message not yet responded to);

Status = unread (message not yet received);

Type = thread (message is part of a thread);

3.9) Generates a unique thread ID (ThreadID) for the thread started by the message.

3.10) Allocates a single record 147 in the threads table 146;

3.11) Updates the thread table record:

ThreadID = thread ID generated by server app. 114;

Subject = subject completed by sender;

SubjectASCI = sort number for subject generated in step;

3.12) Generates a participant ID (ParticipantID) for each participant (1 for sender and 1 for each of the N recipients);

3.13) Allocates in the ThreadParticipants Table 148 N+1 records 149, one for the sender 149s and N for each of the N recipients 149ri (where i is an integer between 1 and N);

3.14) Updates each ThreadParticipant record 149 as follows:

5                   ParticipantID = participant ID generated by server app. 114

                  ThreadID = thread ID generated by server app. 114

                  UserID = message name generated by server app. 114

                  ThreadLevel = 1 (message is at the top of a thread);

10   Referring to FIG. 3B, there is shown an example of how, as a result of the send processing performed by the server application 114, multiple records of the respective tables are generated and cross-referenced. FIG. 3B assumes a simple example where a user with username ArnieS has sent a message to two co-workers, BobB and SueG. Information for these users is provided in the users table 140,  
15   which shows UserIDs (U007, U019, U027) and DisplayNames (Arnie, Robert, SueEllen) for ArnieS, BobB and SueG, respectively. Upon receiving the message the server application 114 creates four new message records M001-M004 in the messages table 142. Two messages (MsgId=M001, MsgId=M002) list ArnieS as MsgRecOwner; one of these has BobB as Recipient and the other has SueG as  
20   Recipient. The other two messages (MsgId=M003, MsgId=M004) are similar, but list BobB and SueG as respective MsgRecOwners. Multiple messages are created so each participant (sender or recipient) can independently manipulate their own messages. For the same message, a single new Threads table 146 record has been allocated (ThreadId = T001). This single record is associated with all  
25   messages having the same subject (i.e., "The Lee Account") and a corresponding unique index (S050). This same ThreadID is associated, for example, with any reply by any of the recipients of the original message as well as any subsequent replies by the original sender. The server application also allocates 3 thread participant records (with ParticipantIds = P021, 025, 032), one each for ArnieS, BobS and  
30   SueG. These participant records map the participants to their respective userIDs (U007, U019, U027) and to the new thread record (ThreadId = T001) associated with the subject common to all messages.

Referring to FIG. 4A, there is shown a data flow diagram illustrating steps performed by the web browser 168-1 of a message sender, the server application 114 and the web browser 168-2 of a message recipient for a message SEND procedure. The steps illustrated in FIG 4A occur after the user of the client 150-1 has sent a message (4.1) designating a user of the client 150-2 as recipient and the server application 114 has processed that message as described in reference to FIG. 3A. As part of the processing described in reference to FIG. 3A the server application 114 determines the recipient(s) of the message (4.2). The server 114 then determines whether the recipient(s) is/are on the system (4.3). If the recipient is on the system, the server application 114 sends a message-alert (MsgAlert) to the recipient (4.4). The MsgAlert (4.4) notifies the intended recipient that a first level message is waiting for him. The recipient application/browser 166-2/168-2 displays the alert in an alert window (4.5) that gives the recipient two response alternatives: OK or CANCEL (4.6). The recipient selects OK to indicate to the server application 114 that he wishes to receive the message. He selects CANCEL to indicate that he does not wish to receive the message and does not want to receive further alerts. The recipient can also choose not to respond to the alert at all. This commonly occurs when the recipient is away from his computer.

20 After sending the MsgAlert (4.1) the server application 114 waits (4.7) for the recipient's response. If the response is NULL, (meaning that the recipient did not respond for some predetermined time interval; e.g., 90 seconds) the server application 114 resends the MsgAlert (4.8a).

25 If the response is OK, the server application sends the Message (4.8b) and updates the messages table 142 to show that the message has been posted. This update involves changing the status 260 of the messages table records 230 for the sender and recipient of the message from "unread" to "read" 230. The server application 114 also sends at the same time any other messages which were in abeyance at the server 100 (i.e., messages previously sent but not OKed for delivery by the recipient). The status of each of these messages is also updated accordingly by the server application 114. The recipient application/browser 166-2, 168-2 then displays the messages in an open, threaded communication board format that is a

key feature of the present invention. This display format is described below in reference to FIG. 4B. A unique aspect of the described sequence of operations is that a user is able to view their communication board messages instantly (that is, as soon as they issue an OK), without first needing to log on to a centralized bulletin board system. Another advantage of the present invention is that the messages are displayed with full threading, which is not possible with conventional instant messaging systems.

If the response is CANCEL, the server application 114 places the message in abeyance (i.e., does not send it) and sends another message to the application/browser 166-2, 168-2 causing a message indication to be displayed/played on the client user interface 158. For example, the message indication could be an alert light or a sound indication, among many other possibilities.

Referring to FIG. 4B, there is shown an image of a communication board format 400 employed by the present invention to display for a single user the contents and status of conversational threads (e.g., 460, 462, 464, 466) in which that single user has participated. This image is displayed as a visual 162 on the user's client computer by the application/browser 166-2/168-2 based on inputs received from the server application 114. The communication board 400 displays all messages and threads owned by a user regardless of subject. These messages are displayed until they are deleted by a participant, they expire, some event occurs that triggers their automatic deletion, or the system administrator deletes them. In addition to messages (including forwarded and carbon copy (cc) messages), the communication board 400 displays announcements broadcast to multiple users and conference notifications.

The communication board 400 lists for each message information from a corresponding messages record, including its MsgTimeStamp 242, Recipient 246, SenderName 245, CcList 250, Subject 254 and MsgText. In the illustrated embodiment the MsgTimeStamp 242, Recipient 246, SenderName 245 and CcList 250 are displayed on the first line 402 of a message, which is called the information

- 22 -

line. The Subject 254 is displayed on the second line 404 and the MsgText 406 is displayed on subsequent lines.

5 In a preferred embodiment, the communication board display is private, meaning that each user can view only their messages (i.e., messages for which a user is listed as MsgOwner in the messages table 142). The present invention is able to support this private treatment of messages as it allocates for each one-to-one communication two message records 230, one owned by the Sender and one by the Recipient. A unique feature of the present invention is that messages and replies  
10 thereto appear simultaneously on the communication boards of both the sender and the recipient.

The client displays the messages with full threading information. That is, first level messages 442 are displayed with no indentation and lower level messages (i.e.,  
15 replies 444 and replies to replies 446) are displayed with corresponding levels of indentation. The information necessary to maintain message threading is provided by the server application 114 from the Parent, Child, ThreadLevel and ThreadId fields 238, 240, 236, 237 of the messages table 142. In particular, each displayed thread comprises a first level message and its children (family of replies). Note that  
20 many threads can have the same subject (e.g., the threads 460 and 464); however, each first level message with the same subject is displayed as a distinct thread.

To assist user recognition of the different message levels and the status of those messages (read, unread, etc.), the displayed embodiment employs color and icons  
25 in addition to indentation. In particular, first level messages are preceded by a filled-in square 408, second level messages (replies) are preceded by a filled-in diamond 410 and third level messages (replies to replies) are preceded by a filled-in circle. In the illustrated embodiment the information line of incoming messages is underlined with different colors depending on whether the message has been  
30 responded to (shown in purple) or need to be responded to (shown in blue). Alternatively, the information line of all incoming messages can be shown in one color (e.g., blue) and with underlining only when the incoming message has not yet been responded to. Note that these display features (indentation, color, icons) are



not required by the present invention but are niceties to assist users in navigating the open, threaded communication board 400.

A novel feature of the present invention is message acknowledge (Ack), which  
5 allows a message recipient to agree to/acknowledge a particular message in such a way that their agreement/acknowledgment is unambiguously recorded by the server application 114. In the displayed embodiment the present invention displays an Ack field 420 alongside the information line 402 of all second and third level messages that indicates when and how (explicitly or implicitly) a message was acknowledged.

10 An Ack field 420 is not displayed next to first level messages in the preferred embodiment, but this could also be done in alternative embodiments. The acknowledge feature is useful in business applications where it can be used to memorialize agreement; e.g., to proposals contained in messages. The acknowledge feature also serves to inform senders as to whether or not recipients  
15 have read their messages.

When a user acknowledges a message (explicitly or implicitly) the server application 114 displays the date and time of the acknowledgment on both the sender and the recipient's communication boards 400. Explicit selection results in the closing of a  
20 thread, meaning that no more replies on that particular thread are possible and that subsequent conversations on the same subject would require a new first level message. In the displayed embodiment the Ack field of closed threads are underlined with a characteristic color (e.g., red) on the communication boards of both participants (e.g., see the Ack field 420-1).

25 A user can implicitly acknowledge a received message by replying to that message (e.g., by sending a third level message). In this case the server application 114 fills in the Ack line 420 of the second level message with the date and time the third level message was sent. When such a reply is sent the server application 114 does  
30 not close the thread as the reply merits its own acknowledgement. As a result, the server application 114 does not display the Ack field 420 with the characteristic color of a closed thread (e.g., see the Ack field 420-2).

If the recipient of a second or third message has not replied or acknowledged to such a message, the Ack field 420 is left blank on the sender's communication board 400 (e.g., see the Ack field 420-3). Therefore, at all times a sender is able to determine the current status of their outgoing messages without needing to login to another service - i.e., the status is displayed through the visual cues presented on the communication board screen 400.

In the preferred embodiment, information lines 402 for incoming messages have a hyperlink function such that selecting the underlined portion of the information line causes the application server 114 to return a reply screen/visual 162 that is partially filled out with the appropriate Recipient, Sender and Subject. The message reply procedures implemented in the present invention are described below, in reference to FIG. 5. There is no hyperlink function associated with the information lines of the outgoing messages.

As an example of these features, refer to FIG. 4B, which shows the communication board 400 for the user, "Mit". The communication board 400 shows Mit has 9 current messages and 4 open messages, which are messages that have not been closed/acknowledged). The underlined message headers are associated with replies to Mit and the plainly-displayed message headers are associated with messages sent by Mit. This example includes four threads 460, 462, 464, 466. The thread 460 comprises two messages of the following levels:

level 1 msg 442, to Arlene from Mit; and

level 2 reply 444 to the level 1 msg;

Each message is displayed in open format, meaning that its text can be read without selecting/opening the message. For example the first level message 442 has the following openly displayed MsgText:

"Pls order 3R report on 145 Piccadilly and give copy to Agnes."

The message 442 was responded to by Arlene, whose reply 444 includes the following MsgText:

"3R report ordered. Will give copy to Agnes."

In response to Arlene's 444, which clearly closed out the conversation, Mit sent an explicit acknowledgement that caused the server application 114 to close the thread 460. Thus, the Ack field 420-1 of the message 444 is shown underlined, indicating thread closure.

5

Due to the symmetry with which the message records are created (i.e, two message records created per communication pair), similar message information is displayed in real time on the communication boards of all participants to a thread. For example, referring to FIG. 4C, there is shown a portion of Arlene's communication board 400A listing some of the same threads 460, 462 as Mit's board 400. In particular, Arlene's thread 460A and messages 442A, 444A correspond to Mit's thread 460 and messages 442, 444. Note that, on both boards 400 and 400A, the Ack information 420-1 for the corresponding messages 444 and 444A is identical. However, the information lines of the messages 444 and 444A differ. That is, on Mit's board, the information line of the message 444 (from Arlene to Mit) is underlined but, on Arlene's board, the corresponding information line is not underlined. This because Arlene was the sender of this message. Other differences between Arlene's and Mit's boards are due to the fact that Arlene and Mit participate in different threads with different users.

20

A user can choose an alternative view of their messages by using the message review board feature of the present invention. A message review board presents a view of all messages in the user's message folder (i.e., stored messages) with a common subject. For example, referring to FIG. 4D, which shows a message review board 1400, the threaded, instant messages shown are all associated with the same subject 1404 ("145 Piccadilly") and are owned by the user, "Mit". The message review board presents the messages in the same manner as a communication board. The underlined message headers are associated with replies to Mit and the plainly-displayed message headers are associated with messages sent by Mit. The example of FIG. 4D includes three threads 1440, 1460, 1480. The thread 1440 comprises three messages of the following levels:

25

30

- level 1 msg 1442, to Arlene from Mit

- level 2 reply 1444 to the level 1 msg; and

a level 3 reply 1446 to the level 2 reply;

Each message is displayed in open format, meaning that its text can be read without selecting/opening the message. For example the first level message 1442 has the  
5 following openly displayed MsgText:

"Where are the signed copies of the TDS and Supp TDS? I need to provided copies to the lender."

The message 1442 was responded to-by Arlene, whose reply 1444 is displayed on  
10 Mit's message review board 1400. Arlene's reply 1444 includes the following MsgText:

"Buyer's agents will be dropping off signed copies tmo. I will go ahead and give copies of same to the Lender when I get them back."

15 Mit replied to the reply 1444 with another reply 1446:

"Thanks a whole lot. That will save me a lot of time. I owe you one!"

By sending this reply 1446 Mit implicitly acknowledged the reply 1444 from Arlene. Consequently, the thread 1440 is still open and the server application 114 sets the  
20 date and time of the acknowledgment 1420-1 to the date and time (10-04-97 16:14) Mit sent the reply 1446.

In response to the reply 1446, which clearly closed out the conversation, Arlene sent an explicit acknowledgement that caused the server application 114 to close  
25 out the thread. Thus, the Ack field 1420-2 of the message 1446 is shown underlined, indicating thread closure.

The preceding description is directed to an embodiment of the present invention where the communication boards are private and provide instant, open display of  
30 threaded messages. Alternative and equally novel embodiments of communication systems can employ various combinations of these concepts (privacy, instant messaging, threading and open display). For example, the teachings of the present invention could be employed in the following novel systems:

- 1) public bulletin boards with open display of messages;
- 2) e-mail systems with open display of messages;
- 3) e-mail systems with threading of messages;
- 4) private bulletin boards with no other unique features;
- 5) private bulletin boards with instant messaging;
- 6) private bulleting boards with instant messaging and open display; and
- 7) instant messaging systems with threaded of messages.

Descriptions of these embodiments are not provided herein as their respective implementations should be apparent from the descriptions already provided. Other embodiments consistent with these teachings and descriptions will occur to the reader skilled in the art and are within the scope of the present invention. Having described the communication board concept, the message reply process is now described.

Referring to FIG. 5A, there is shown a data flow diagram illustrating steps performed by a web browser 168-1 of a message replier, the server application 114, and a web browser 168-2 of a message reply recipient for a message reply procedure. The steps illustrated in FIG 4A occur after the user of the client 150-2 has received a message (4.1) from a user of the client 150-1. Upon displaying a message on their communication board as described in reference to FIG. 4, a user can choose to reply to that message by selecting a reply option from a menu, an icon, or other equivalent methods (5.1). Once the user has selected the reply option a reply box is displayed (5.2) that allows the user to specify the reply's recipient and subject (these fields are filled in by default with the sender's information) and MsgText. The reply box description could be sent by the server application 114 or could be generated by the client application 166-1. The user fills in the reply (5.3), and then sends it off to the server application (5.4). Key information conveyed to the server in the reply includes the MsgID of the message responded to.

In response, the server application 114 assigns a unique MsgID to the reply (5.5), generates corresponding message records 230 for the reply sender and recipient (5.6) and updates database 108 records accordingly, including setting parent and

child pointers to and from other message records 230 (5.7). The server application 114 then posts its reply to the indicated recipients (e.g., the user of the client 150-1) IF they are online (5.8). Note that, unlike first level messages, the server application 114 does not issue queries to recipients asking if they would like to a  
5 reply to be sent. Instead, the server application 114 pushes replies to recipients. Moreover, every time it pushes one reply, the server application 114 pushes all other replies held in abeyance (except for first level messages not yet accepted). An illustration of the databases 108 reflecting processing by the server application 114 in response to a reply is now described in reference to FIG. 5B.

10 Referring to FIG. 5B, there is shown an example of how, as a result of the reply processing performed by the server application 114, multiple records of the respective tables are generated and cross-referenced. FIG. 5B assumes a simple example where SueG has responded to the message sent by ArnieS (refer to  
15 discussion of FIG. 3B). Upon receiving the reply message the server application 114 creates two new message records M005, M006 in the messages table 142. One of these messages (MsgId=M005) lists ArnieS as MsgRecOwner and Recipient and SueG as Sender. The other message (MsgId=M006) is similar, but lists SueG as MsgRecOwner and sender and ArnieS as Recipient. A new Threads table 146  
20 entry has not been allocated as the subject (i.e., "The Lee Account") and subject index (S050) for the reply are already represented by the Thread T001. Nor are new ThreadParticipant table 148 entries allocated. This is because the two participants in the reply (ArnieS and SueG) have appropriate records (with ParticipantIds = P021 and 032) in the ThreadParticipants table 148.

25 Referring to FIG. 6, there is shown a data flow diagram illustrating steps performed by the web browser 168-1 of a message acknowledger and the server application 114 for a message acknowledge procedure. The steps illustrated in FIG 4A occur after the user of the client 150-2 has received a reply message (5.7) from a user of  
30 the client 150-1. A user can choose to acknowledge explicitly a message displayed on their communications board 400 by selecting an Ack option from a menu, icon, or other equivalent means (6.1). In response, the client application/browser 166-2/166-3 relays pertinent information (including MsgId) for the message being explicitly

acknowledged to the server application 114 (6.2). The server application 114 processes the message acknowledgment as described in reference to FIGS. 4A-4C and updates the databases 108 accordingly (6.3), principally by changing the status of the relevant message record to "ACKed" and completing the AckDate 256 (FIG. 2). The server application 114 then posts the acknowledgement to the respective communications boards 400 of both participants (i.e., the sender using the client 150-1 and the recipient using the client 150-2) as described in reference to FIGS. 4B-4C (6.4). In a preferred embodiment the acknowledgment is only sent when the sender is online.

Alternatively, as described above, an acknowledgment can be sent implicitly as the result of a recipient of other than a first level message responding to that message. In this situation the server application 114 allocates new message records 230 in the same manner as for a reply (FIG. 5B); i.e., the Status 260 of those records is not listed as Acked and the AckDate 256 (FIG. 2) is not filled in. Additionally, the IsRespondedTo flag of the parent messages is set.

An example of the databases 108 resulting from the processing of an acknowledgment is not shown given the similarity of these results to those in the reply case, already described in reference to FIG. 5B.

In addition to the communication board features described above, the present invention provides additional **conference, whisper, talk and mail** modes. It is a key feature of the present invention that these modes are integrated with the communications board features. It is now described reference to FIG. 7 how the server application 114 supports **conference** mode.

Referring to FIG. 7A, there is shown a flow chart illustrating steps by which the server application 114 schedules a conference, notifies conference participants and holds the conference. As the first step in scheduling a conference a user/moderator begins conference mode operations (702). The moderator then schedules a conference (704) by defining its:

**participants** (one or more users if conference is private, not necessary when conference is open);

**topic;**

**data and time;** and

5 **type** (open or private), where open means that any user can participate in the conference and private means that a user can only participate if invited by the moderator.

This information is entered by the moderator on a conferences page 162 generated  
10 by a browser 168 from information (typically formatted as ACTIVE SERVER PAGES) forwarded by the server application 114. The completed conference page 162 is returned to the server application 114, which allocates a new record 270 in the conferences table 144 (706). The server application 114 updates the fields (706) of the new conference record 270 as follows:

15	<b>ID</b>	set to a unique ID generated by the application 114;
	<b>Moderator</b>	set to the user name of the moderator;
	<b>Topic</b>	set the topic defined by the user on the conferences page 162;
	<b>Type</b>	set to type (open or private) selected by the moderator;
20	<b>Participants</b>	set to participants entered by the moderator;
	<b>IsScheduled</b>	set to 1 if the user indicated that the conference is to be scheduled for future as opposed immediately; and
	<b>When</b>	date and time entered by moderator.

Finally, the server application 114 schedules a virtual conference room for the  
25 future, or immediately, depending on when the conference is scheduled (706).

Depending on when the conference is scheduled, at an appropriate time (which could be immediately or sometime in the future) (708-Y) the server application 114 sends notifications to the conference participants (710).

30

As shown in FIG. 7B, which is an image of user Mit's communication board 720 including conference notifications and announcements, the notifications are displayed similarly to messages (including the possibility of acknowledgements). In



particular, open conference notifications, such as the notifications 722, 728, are sent to all users, and private conference notifications, such as the notifications 746, 752, and 756, are only sent to invited participants. Referring to an exemplary notification 722, all notifications display:

- 5           a subject 740 that indicates the type of notification (e.g., PRIVATE CONFERENCE NOTIFICATION);
- the virtual conference room 742 (e.g., Room 1258);
- the date and time 744 of the conference (e.g., NOW);
- the topic 746 (e.g., "Affect of new Health Ins Benefits"); and
- 10          —an IN PROGRESS indication 748 if the conference is in progress.

Note that private conference notifications can be acknowledged in the same manner as messages. For example, in response to the notification 732 from Arlene regarding a private conference to discuss future Christmas parties, Mit issued a  
15       confirming reply 734, which was later acknowledged 750 by Arlene. The acknowledgment 750 also closes the thread consisting of the messages 732, 734.

FIG. 7B shows another feature of the present invention, announcements 724, 730, which are immediate messages broadcast to all users who are online.

20

Referring again to FIG. 7A, at the scheduled time the server application 114 hosts the conference (714) by:

- allocating the virtual conference room;
- notifying all participants again that the conference is starting (e.g., the IN  
25       PROGRESS notification 722);
- and allowing the moderator to administer the conference in accordance with the type of conference (e.g., if the conference is public, users can join at will, but if the conference is private, users can only join at if invited by the moderator).

30

One example of a conference session screen 770 is shown in FIG. 7C. The screen 770 includes an agenda 772 defined by the moderator, a comment screen 774 on which comments typed by participants on their own conference input screens are

displayed and a list of participants 776. This screen 770 is displayed for all users who are participants in the conference by their respective browsers 168. In a preferred embodiment all conference information is logged by the application server 114 on the hard disk 104. The screen 770 includes an ANON button, which, when  
5 selected, allows a user to participate in the conference anonymously.

A unique feature of the present invention is **whisper mode**, which is implemented in conjunction with **conference mode**. During a conference two users who wish to carry on a private, unlogged conversation can enter whisper mode. When in  
10 whisper mode users view their conversation on a whisper mode screen on which only their comments are displayed. No other user can view comments made by whisper mode participants.

Referring to FIG. 8, there is shown an exemplary talk session dialog input screen  
15 800 in which a user (e.g., Arlene) enters a talk message 804 she would like to send to another user 802 (e.g., Mit) as part of a talk session. In the preferred embodiment, a user can start a talk session from any system mode other than conference mode. When the user is satisfied with the message 804, they select the send button 806, which causes the client application/browser 166, 168 to send the  
20 information from the talk session dialog input screen to the server application 114.

The server application 114 then determines whether the intended talk session participant (e.g., Mit is online). If the intended participant is online, the server application 114 sends him a talk mode incoming message box, which announces a  
25 new incoming message and gives the intended participant the option of checking the message (i.e., joining in the talk session) or declining to participate. The incoming message box is displayed for the intended recipient no matter which mode his client 150 is in. If the intended participant declines to talk or is not available, the server application 114 sends the requestor (e.g., Arlene) a party not available  
30 message, indicating that the talk session cannot commence. If the intended participant is available and agrees to join the talk session, the server application 114 causes his client application/browser 166/168 to display the message and gives him an opportunity to respond.

If the user elects to respond, he enters a message in a response input box and causes the appropriate browser 168 to send the information defined therein to the server application 114. The response input box is very similar to the talk session dialog input screen 800, shown in FIG. 8 and is therefore not described herein.

5 Upon receiving the response, the server application 114 resends the first participant (e.g., Arlene) a dialog screen showing the second participant's response alongside the first participant's initial message. From this screen the first participant has the opportunity to send a reply back to the second participant. An exemplary talk dialog screen 820 for Arlene is shown in FIG. 8B. This screen shows in its upper section  
10 822 Arlene's initial message 824 and Mit's response 826. The screen 820 also includes a response input box 830 in which Arlene can enter a subsequent response. The talk session can proceed with screens like the screen 820 until one of the users signals an intention to exit the talk mode.

15 As an example of the integration provided by the present invention, the server application 114 causes a incoming message alert to pop up during talk mode to notify a talk mode participant that he has a waiting communication board message. The alerted user can choose to check the message or CANCEL the alert, causing the server application 114 to initiate message processing operations already  
20 described in reference to FIGS. 4-6. While the user is checking his messages, the server application 114 causes the browser 168 used by the other talk session participant to display a talk session paused message. The talk session is re-activated when the participant checking his messages chooses to rejoin the talk session.

25 Talk session information is centrally maintained - in this case in a talk table 850 (stored in the database 108). Unlike most other modes (except for whisper mode), talk messages are not logged on the hard disk 104 server; therefore, the talk table 850 only includes a small set of status information for each talk session. In a  
30 preferred embodiment this status information includes:

<b>TalkSessID</b>	Unique key for each talk session generated by the server;
<b>FirstParticipant</b>	First participant user name;

- 34 -

	<b>ScndParticipant</b>	Second participant user name;
	<b>InActive</b>	Set to indicate that one of participants has not yet agreed to join session;
	<b>MessageFlg</b>	Set to indicate that one of the participants has paused session to check a new message;
5.	<b>MessageTS</b>	Date and time when participant paused session to check new message.

The server application 114 allocates a single talk record for each new talk session.

- 10 There is no need for the server application 114 to keep track of individual responses as talk sessions are not logged in the preferred embodiment.

- Another mode offered by the present invention is mail mode. Unlike conventional e-mail programs, in its mail mode the present invention is able to provide threaded  
 15 mail over the Internet. An example of a mail screen 900 displayed for a particular user is shown in FIG. 9.

- The mail screen 900 lists mail received 902 and sent 904 by a particular user (e.g., Mit). Each incoming e-mail message is treated by the server application 114 as a  
 20 level one message that begins a respective e-mail thread. Replies to the incoming e-mail messages are indented on the mail screen 900, visually indicating their position as second level messages in their associated thread. For example, the incoming e-mail messages 906, 908 and 910 all have replies 906a, 908b, 910c.
- 25 Each of the incoming messages has a subject line (e.g., the subject 912 of the message 906) that is underlined. When a user selects the underlined subject line he prompts the server application 114 to return to the user's client application/browser 166/168 an e-mail reply box with most fields (such as sender, recipient, subject, date and time) already filled-in. The user then fills in the message  
 30 text and sends the e-mail message. As with the other modes, after a reply is sent the mail screen 902 is immediately updated by the server application 114 with the threading information. Outgoing mail is handled in the same manner except for the

problem that threading information may not be maintained by external e-mail servers that receive the outgoing messages.

As with the other system modes, e-mail information is centrally maintained - in this case in a mail table 920 (stored in the database 108) whose fields include:

5	<b>MailID</b>	Unique key for each email message generated by the server;
	<b>MailTimeStamp</b>	Date and time message was sent or received;
	<b>Sender</b>	Sender Name;
10	<b>SenderAdr</b>	Sender e-mail address;
	<b>Recipient</b>	Recipient Name;
	<b>RecipientAdr</b>	Recipient e-mail address;
	<b>ThreadId</b>	Unique Id for each mail Thread;
15	<b>ThreadLevel</b>	Top level messages have level = 1, Replies have level = 2;
	<b>Subject</b>	Message subject; and
	<b>MsgFN</b>	Name of file on hard disk 104 where MsgText is stored.

The server application 114 allocates a mail message record for each new e-mail (outgoing, incoming, or reply) and updates these e-mail fields in much the same way as in the message situation described above. As a result, the processing of e-mail messages by the present invention is not further described.

While the present invention has been described with reference to a few specific embodiments, the description is illustrative of the invention and is not to be construed as limiting the invention. Various modifications may occur to those skilled in the art without departing from the true spirit and scope of the invention as defined by the appended claims.

## WHAT IS CLAIMED IS:

1. A system for threaded instant messaging for use in a network of computers including a server computer, comprising:

5 a server mechanism executable in the server configured to make instantly available to a user who is logged onto the server representations of messages sent to the user and to maintain threading information associated with the messages;

the server mechanism also being configured to display the representations of messages to the user along with a representation of the threading information.

10 2. The system of claim 1, wherein the server mechanism is configured to display the representations in an open format wherein the messages do not need to be selected to be read.

15 3. The system of claim 1, wherein the server mechanism is configured to display the representations on a private message board that only shows communications sent to and by the user and which can be viewed only by the user.

20 4. The system of claim 3, wherein the server is configured to maintain a plurality of private message boards, one for each user of the system; such that, whenever a representation of the message is displayed on the private message board of one participant in the message, a corresponding representation of the message is displayed nearly simultaneously on the private message board of another participant in the message.

25 5. The system of claim 1, wherein the server mechanism is configured to enable users to acknowledge messages; such that, when the user acknowledges a particular message, the server mechanism notifies a second user who is a participant in the message of the user's acknowledgment of the particular message.

30 6. The system of claim 5, wherein the acknowledgment can be explicit or implicit, such that:

when the acknowledgment is explicit, the server mechanism closes a thread including the particular message; and

when the acknowledgment is implicit, the server mechanism leaves open the thread including the particular message, the implicit acknowledgment occurring  
5 whenever the user replies to the particular message.

7. The system of claim 6, wherein the explicit acknowledgment is employed by the user to manifest acceptance of/agreement to contents of the message.

10 8. The system of claim 1, wherein the server mechanism is configured to queue at the server at least a subset of the messages addressed to the user without displaying to the user the representations of the subset until the user has approved transmission of at least one of the messages in the subset, after which the server mechanism displays the representations of the subset to the user.

15 9. The system of claim 1, wherein the server mechanism displays the representation of the message along with a hyperlink field, the selection of which by the user causes the server mechanism to display to the user a reply window with at least some fields filled out with information associated with the hypertext field,  
20 enabling the user to formulate a reply to the sender of the message.

10. The system of claim 1, further comprising:

a message table including message records, each of which is associated with a message having a sender, recipient, owner and subject;

25 such that, whenever the user sends a message to N other users, the server mechanism allocates  $2 \times N$  message records, N of which have respective ones of the other users as the owner and the other N of which have the user as the owner, all of the  $2 \times N$  messages having the same sender and subject, respective pairs of the  $2 \times N$  messages listing as the recipient a respective one of the other users;

30 the server mechanism being additionally configured to display for each of the N + 1 users only those message for which he is owner.

11. The system of claim 10, wherein the server mechanism enables each of the users to manipulate only those messages for which he is the owner.

12. The system of claim 11, wherein messages with the same subject, sender, recipient and message text are displayed nearly simultaneously to the respective owners of the messages.

13. A system for open display bulletin boards for use in a network of computers including a server computer, comprising:

10 a server mechanism executable in the server configured to provide a bulletin board system wherein messages submitted to the bulletin board by users of the network are displayed with threading information and in an open format wherein all messages are displayed in full and do not require prior selection for viewing.

15 14. The system of claim 13 wherein the server mechanism is configured to display the messages on a plurality of private message boards, each of which only shows communications sent to and by a respective user and which can be viewed only by the respective user.

20 15. The system of claim 14, wherein, whenever a first representation of a particular message is displayed on the private message board of one participant in the particular message, a corresponding representation of the particular message is displayed nearly simultaneously on the private message board of another participant in the particular message.

25

16. The system of claim 13, wherein the server mechanism is configured to enable users to acknowledge messages; such that, when the user acknowledges a particular message, the server mechanism notifies a second user who is a participant in the particular message of the user's acknowledgment of the particular message.

30

17. The system of claim 16, wherein the acknowledgment can be explicit or implicit, such that:



when the acknowledgment is explicit, the server mechanism closes a thread including the particular message; and

when the acknowledgment is implicit, the server mechanism leaves open the thread including the particular message, the implicit acknowledgment occurring  
5 whenever the user replies to the particular message.

18. The system of claim 17, wherein the explicit acknowledgment is employed by the user to manifest acceptance of/agreement to contents of the particular message.

10 19. The system of claim-13, wherein the server mechanism is configured to queue at the server at least a subset of the messages addressed to the user without displaying the representations of the subset until the user has approved transmission, after which the server mechanism displays the representations of the subset to the user.

15

20. The system of claim 13, wherein the server mechanism displays the representation of the message along with a hyperlink field, the selection of which by the user causes the server mechanism to display to the user a reply window with at least some fields filled out with information associated with the hypertext field,  
20 enabling the user to formulate a reply to the sender of the message.

21. The system of claim 13, further comprising:

a message table including message records, each of which is associated with a message having a sender, recipient, owner and subject;

25 such that, whenever the user sends a message to N other users, the server mechanism allocates  $2 \times N$  message records, N of which have respective ones of the other users as the owner and the other N of which have the user as the owner, all of the  $2 \times N$  messages having the same sender and subject, respective pairs of the  $2 \times N$  messages listing as the recipient a respective one of the other users;

30 the server mechanism being additionally configured to display for each of the N + 1 users only those message for which he is owner.

22. The system of claim 21, wherein the server mechanism enables each of the users to manipulate only messages of which he is the owner.

23. The system of claim 22, wherein messages with the same subject, sender,  
5 recipient and message text are displayed nearly simultaneously to the respective owners of the messages.

24. A private message board system for use in a network of computers including a server computer, comprising:

10 a server mechanism executable in the server configured to provide a plurality of private bulletin boards for each of a plurality of users of the system, wherein each of said private bulletin boards shows history of messages associated with conversations involving a respective user.

15 25. The system of claim 24, wherein the server mechanism is configured to display representations of the messages in an open format wherein the messages do not need to be selected to be read.

20 26. The system of claim 24, wherein the server mechanism is configured to display nearly simultaneously with its posting a particular message on the private message boards of all participants in the message.

27. The system of claim 24, wherein the server mechanism is configured to enable users to acknowledge messages; such that, when the user acknowledges a  
25 particular message, the server mechanism notifies a second user who is a participant in a thread containing the thread of the user's acknowledgment of the particular message.

28. The system of claim 27, wherein the acknowledgment can be explicit or  
30 implicit, such that:

when the acknowledgment is explicit, the server mechanism closes the thread including the particular message; and

when the acknowledgment is implicit, the server mechanism leaves open the thread including the particular message, the implicit acknowledgment occurring whenever the user replies to the particular message.

- 5     29.     The system of claim 24, wherein the server mechanism is configured to queue at the server at least a subset of the messages addressed to the user without displaying the representations of the subset until the user has approved transmission, after which the server mechanism displays the representations of the subset to the user.

10

30.     The system of claim 24, wherein the server mechanism displays a representation of the message along with a hyperlink field, the selection of which by the user causes the server mechanism to display to the user a reply window with at least some fields filled out with information associated with the hypertext field,
- 15     enabling the user to formulate a reply to the sender of the message.

31.     The system of claim 24, further comprising:  
a message table including message records, each of which is associated with a message having a sender, recipient, owner and subject;

20

- such that, whenever the user sends a message to N other users, the server mechanism allocates  $2 \times N$  message records, N of which have respective ones of the other users as the owner and the other N of which have the user as the owner, all of the  $2 \times N$  messages having the same sender and subject, respective pairs of the  $2 \times N$  messages listing as the recipient a respective one of the other users.

25

32.     The system of claim 31, wherein messages with the same subject, sender, recipient and message text are displayed nearly simultaneously on the private message boards of the respective owners of the messages.

- 30     33.     A communication board system for use in a computer network including a server, comprising:

a server application executable on the server; and

a client application providing an interface between system users and the server application;

such that said client application and server application are configured cooperatively to provide each of said users with a respective view of a virtual  
5 bulletin board system illustrating history and content of at least a subset of said messages, said respective view being instantly accessible to and customizable for a corresponding one of said users.

34. The communication board system of claim 33, wherein the server application  
10 and the client application are browser-based, enabling the communication board system to be implemented on any computer network compatible with Internet-based protocols.

35. The communication board system of claim 34, wherein the computer network  
15 is selected from:  
the Internet;  
an intranet; or  
an extranet.

20 36. The communication board system of claim 33, wherein said respective view includes only those of said messages associated with conversations to which said corresponding user is a participant.

37. The communication board system of claim 33, further comprising:  
25 a data repository in which the server application stores information about system users and messages exchanged between said users.

38. The communication board system of claim 37, wherein the data repository  
30 comprises:  
a message table including message records, each of which is associated with a message having a sender, recipient, owner, subject and thread id;  
such that, whenever a user sends a message to N other users, the server application allocates a message family of  $2 \times N$  message records, N of which have

respective ones of the other users as the owner and the other N of which have the first user as the owner, all of the  $2 \times N$  messages having the same sender, subject and thread id, respective pairs of the  $2 \times N$  messages listing as the recipient a respective one of the other users, the sending user and other users being thread participants.

39. The communication board system of claim 36, wherein messages with identical subject, sender, recipient and thread id can be provided nearly simultaneously on the respective views provided to the respective owners of the messages.

40. The communication board system of claim 38, wherein, when a first user sends a first level message to a second set of users using the client application, the server application is configured in response to:

allocate the family (first level family) of message records for the first level message;

generate a unique thread id and associate the unique thread id with each of the first level family;

update each member of the first level family with the sender, recipient, owner and subject information for a respective thread participant;

issue a message alert to the client applications of each of the other users;

transmit a representation of a respective message record to the client application of each of the second set of users who agrees to delivery of the message; and

hold in abeyance transmission of a representation of a respective message record to each of the second set of users who does not agree to delivery of the message.

41. The system of claim 40, wherein, when a third user sends a reply to a message from a fourth user using the client application, the server application is configured in response to:

allocate the family (reply family) of message records for the reply message;

associate the unique thread id from a parent family with each member of the reply family, the parent family being the message records associated with the message from the fourth user;

update each member of the reply family with the sender, recipient, owner and subject information for a respective participant in the reply message;

link members of the reply family with related message records of the parent family; and

transmit a representation of a respective reply message record to the client application of the fourth user.

10

42. The communication system of claim 41, wherein, when a third user explicitly acknowledges an incoming message using the client application, the server application is configured in response to:

update the message records associated with the thread that includes the incoming message to show that the thread is closed;

transmit a representation of the incoming message to the client applications of participants in the thread that includes the incoming message conveying that the thread is closed.

43. The communication system of claim 41, wherein, when a third user implicitly acknowledges an incoming message using the client application, the server application is configured in response to:

update the message records associated with the thread that includes the incoming message to show that the thread is responded to;

transmit a representation of the incoming message to the client applications of participants in the thread that includes the incoming message conveying that the thread is responded to.

25

44. The communication board system of claim 36, wherein the respective views display representations of the messages in an open format wherein the content of the messages can be read without selecting the messages.

30

45. The communication board system of claim 36, wherein the respective views display the representation of a message along with a hyperlink field, the selection of

which by the user causes the server and client applications cooperatively to display to the user a reply window with at least some fields filled out with information associated with the hypertext field, enabling the user to formulate a reply to the message.

5

46. A method for threaded instant messaging for use in a computer network, comprising the steps of:

displaying messages sent over the network involving a user of the network so that only said user can view messages sent and received by said user;

10 displaying said messages in an open format so that content of said messages is viewable without selection; and

immediately making said messages available for viewing by said user whenever said user is online.

15

47. The method of claim 46, further comprising the steps of:

maintain threading information of said messages; and

displaying a representation of said threading information along with said messages.

20

48. The method of claim 46, further comprising the steps of:

whenever a user sends a message to N other users, allocating a message family of  $2 \times N$  message records, N of which have respective ones of the other users as the owner and the other N of which have the first user as the owner, all of the  $2 \times N$  messages having the same sender, subject and thread id, respective pairs of the  
25  $2 \times N$  messages listing as the recipient a respective one of the other users, the sending user and other users being thread participants.

49. The communication board system of claim 48, further comprising the steps of: when a first user sends a first level message to a second set of users:

30

allocating the family (first level family) of message records for the first level message;

generating a unique thread id and associate the unique thread id with each of the first level family;

updating each member of the first level family with the sender,  
recipient, owner and subject information for a respective thread participant;  
issuing a message alert to each of the other users;  
transmitting a representation of a respective message record to each  
5 of the second set of users who agrees to delivery of the message; and  
holding in abeyance transmission of a representation of a respective  
message record to each of the second set of users who does not agree to delivery  
of the message.

10 50. The method of claim 49, further comprising the steps of:  
when a third user sends a reply to a message from a fourth user,  
allocating the family (reply family) of message records for the reply  
message;  
associating the unique thread id from a parent family with each  
15 member of the reply family, the parent family being the message records associated  
with the message from the fourth user;  
updating each member of the reply family with the sender, recipient,  
owner and subject information for a respective participant in the reply message;  
linking members of the reply family with related message records of  
20 the parent family; and  
transmitting a representation of a respective reply message record to  
the fourth user.

25 51. The method of claim 50, further comprising the steps of:  
when a third user explicitly acknowledges an incoming message,  
updating the message records associated with the thread that includes  
the incoming message to show that the thread is closed; and  
transmitting a representation of the incoming message to participants  
in the thread that includes the incoming message conveying that the thread is  
30 closed.

52. The method of claim 50, further comprising the steps of:  
when a third user implicitly acknowledges an incoming message,



updating the message records associated with the thread that includes the incoming message to show that the thread is responded to; and

transmitting a representation of the incoming message to the client applications of participants in the thread that includes the incoming message conveying that the thread is responded to...

5

53. A system for memorializing agreement for use in conversations and negotiations conducted via an electronic communication system, comprising:

a server program that opens a thread corresponding to each conversation ongoing in the communication system;

10

a data repository in which the server program records status and content of messages composing said conversation; and

a client program configured to cooperate with said server program to enable users of the electronic communication system to view and respond to said

15

messages;

a user response including acknowledgment, wherein a user signifies agreement to a particular message by acknowledging said particular message, said client program being configured to relay said acknowledgment to said server program, which is configured to close said particular message's thread and update said data repository to show said status of said particular message is acknowledged.

20

54. A threaded e-mail system for use in a computer network, comprising:

a server application running in a server attached to the computer network; an e-mail database;

25

a client application employed by users to interact with the server application to perform e-mail operations, including exchanging e-mail with external correspondents and viewing e-mail messages and e-mail status;

the server application being configured to allocate a first new record in the e-mail database corresponding to each new incoming e-mail and to associate each of the first new records with a unique e-mail thread id;

30

the server application being configured to allocate a second new record in the e-mail database corresponding to each reply to the incoming e-mail and to

associate each of the second new records with the unique e-mail thread id of the incoming e-mail replied to; and

- 5       the server application being configured to present information to the client application from the e-mail database enabling the client to display the e-mail messages and e-mail status including threading information showing common subject matter and history of the incoming e-mail and the replies thereto issued by a respective user.

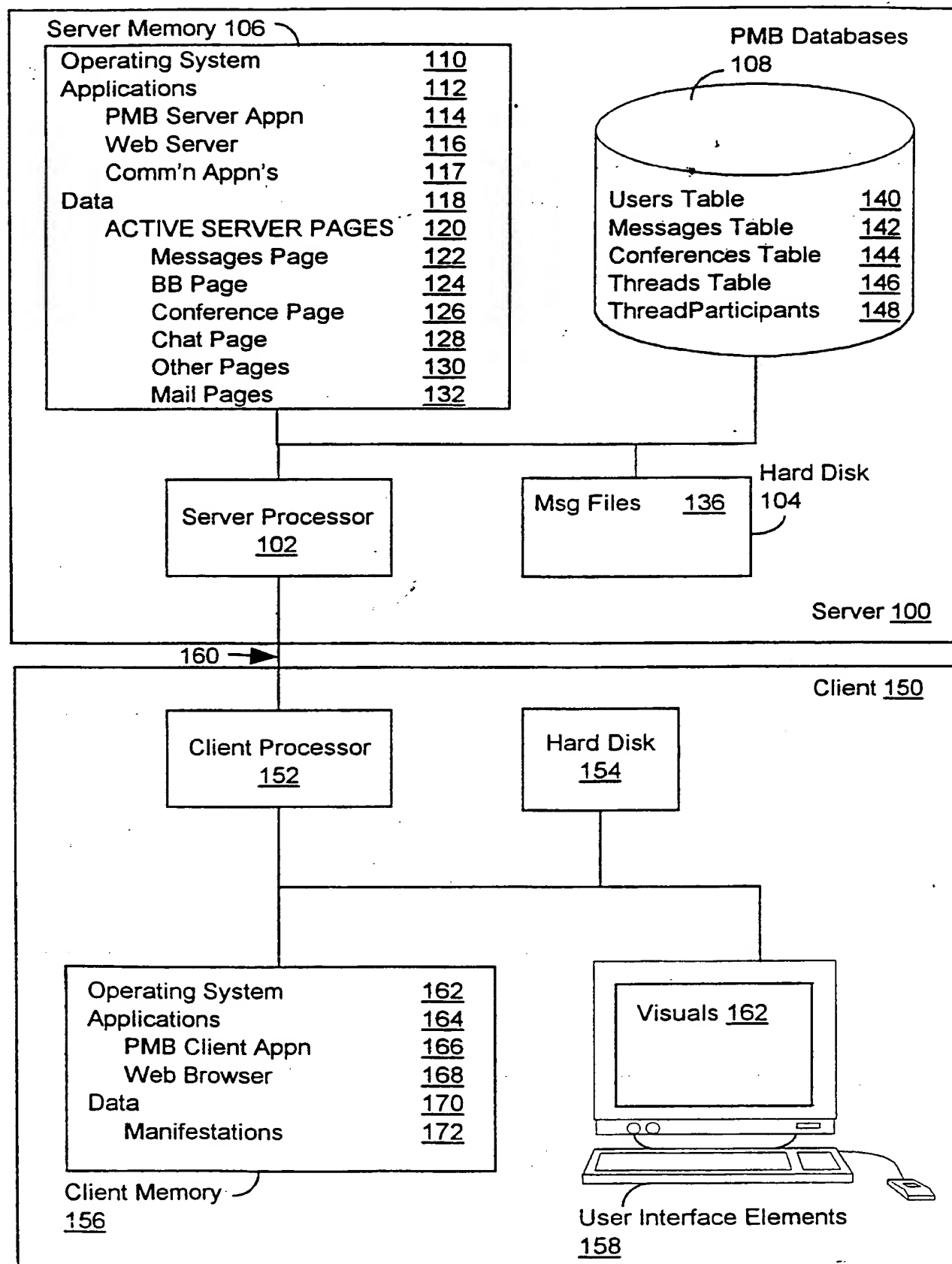


FIG. 1

Users Table

140

User i	<u>200</u>
ID	<u>202</u>
Name	<u>204</u>
Displayname	<u>206</u>
Password	<u>208</u>
HasNewMail	<u>210</u>
HasInvitation	<u>212</u>
HoldAlerts	<u>214</u>
Preferences	<u>218</u>

Conferences Table

144

Conference k	<u>270</u>
ID	<u>272</u>
Moderator	<u>274</u>
Topic	<u>276</u>
Type	<u>278</u>
Participants	<u>280</u>
IsScheduled	<u>282</u>
When	<u>284</u>

Threads Table 146

Thread m	
ThreadID	<u>286</u>
Subject	<u>288</u>
Subject ASCII	<u>290</u>

Messages Table

142

Message j	<u>230</u>
MsgID	<u>232</u>
MsgFN	<u>234</u>
ThreadLevel	<u>236</u>
ThreadID	<u>237</u>
ParentID	<u>238</u>
ChildID	<u>240</u>
MsgTimeStamp	<u>242</u>
MsgOwner	<u>243</u>
SenderId	<u>244</u>
SenderName	<u>245</u>
Recipient	<u>246</u>
RecipientList	<u>248</u>
CcList	<u>250</u>
IsCarbonCopy	<u>252</u>
Subject	<u>254</u>
SubjectASCII	<u>255</u>
AckDate	<u>256</u>
IsRespondedTo	<u>258</u>
Status	<u>260</u>
Type	<u>262</u>
ExpiryDate	<u>264</u>
IsForward	<u>266</u>

Thread Participants 148

Thread Participant n	
ParticipantID	<u>286</u>
ThreadID	<u>288</u>
UserID	<u>290</u>

Unread  
Read  
Responded  
Acked  
Stored  
Deleted

Thread  
Invitation  
Confirmation  
Log

FIG. 2

Threads Table 146

Thread Rec 1  
 ThreadID = T001  
 Subject = Lee Account  
 SubjectASCI = S050

Messages Table 142

Message Rec1  
 MsgID = M001  
 MsgFN = M001.htm  
 ThreadId = T001  
 MsgRecOwner = ArnieS  
 SenderId = U007  
 SenderName = ArnieS  
 Recipient = BobB  
 Recipient List =  
     BobB, SueG  
 Subject = Lee Account  
 SubjectASCI = S050  
 ...

Message Rec2  
 MsgID = M002  
 MsgRecOwner = ArnieS  
 Recipient = SueG

Message Rec3  
 MsgID = M003  
 MsgRecOwner = BobB  
 Recipient = BobB  
 ...

Message Rec4  
 Msg = M004  
 MsgRecOwner = SueG  
 Recipient = SueG  
 ...

Threads Participants Table 148

Thread Participant Rec 21  
 ParticipantId = P021  
 ThreadID = T001  
 UserID = U007  
 ...

Thread Participant Rec 25  
 ParticipantId = P025  
 ThreadID = T001  
 UserID = U019  
 ...

Thread Participant Rec 32  
 UserID = U027  
 ...

Users Table 140

User Rec 7  
 UserId = U007  
 Name = ArnieS  
 DisplayName = Arnie  
 HoldAlerts = NO

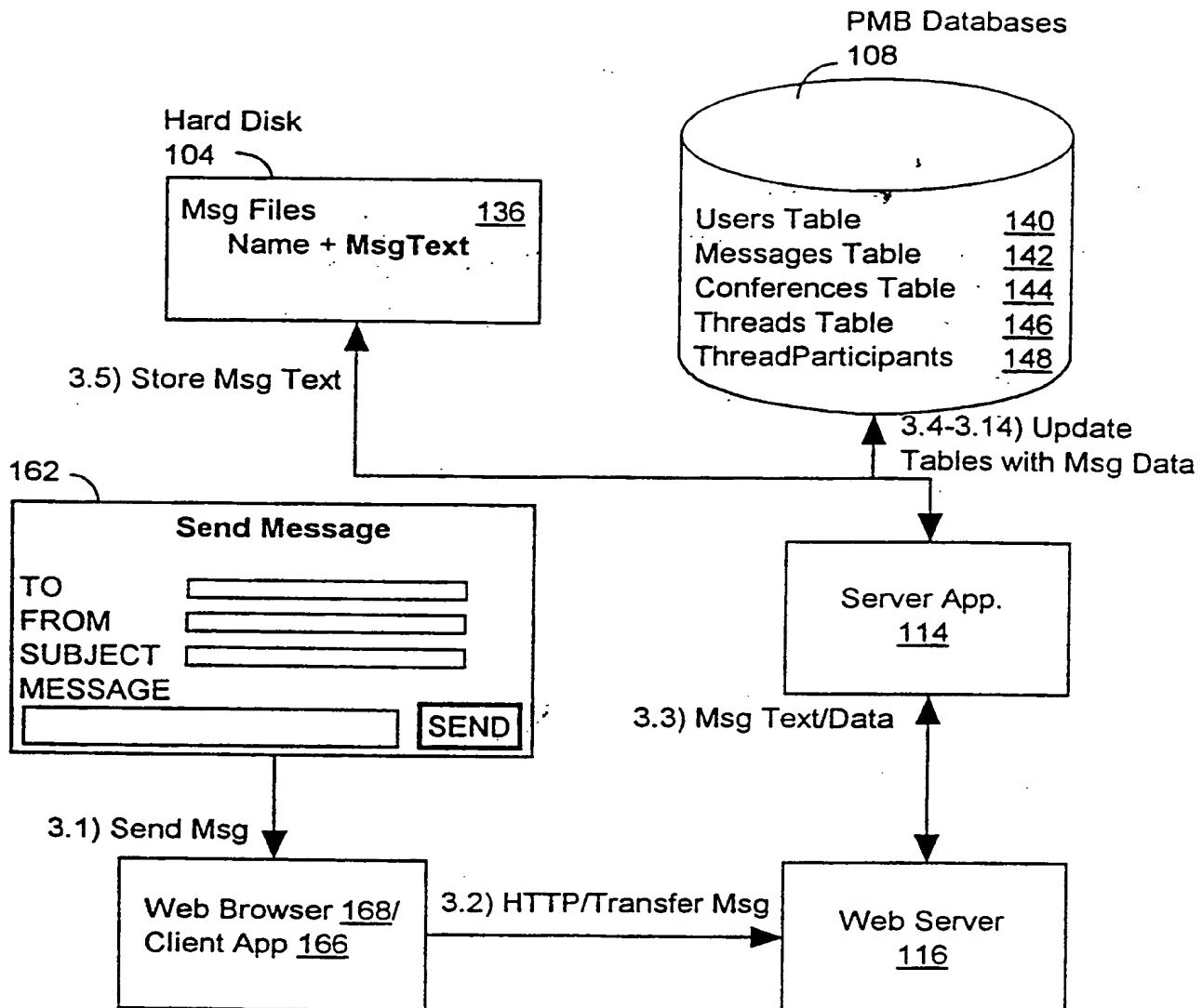
User Rec 19  
 UserId = U019  
 Name = BobB  
 DisplayName = Robert  
 HoldAlerts = YES

User Rec 27  
 UserId = U027  
 Name = SueG  
 DisplayName = SueEllen  
 HoldAlerts = NO

Hard Disk 104

Msg Files	<u>136</u>
M001.htm:	
MsgTxt	

FIG. 3B



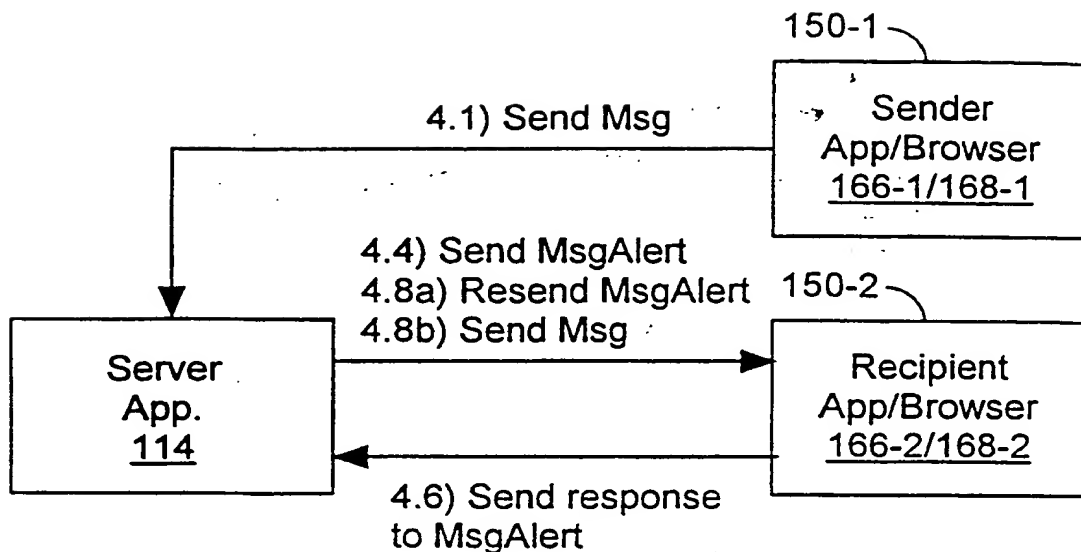
3.1)  
 SendMsg(  
   TO: Recipient  
   FROM: Sender  
   SUBJECT: Subject  
   MESSAGE: MsgText)

3.2)  
 Transfer(  
   URL: SendMsg(  
     Recipient,  
     Sender,  
     Subject,  
     MsgText))

3.3)  
 MsgData(  
   TO: RecipientName  
   FROM: SenderID  
   SUBJECT: Subject  
   MESSAGE: MsgText)

3.4 to 3.14) Update Tables(  
   MsgID, Name, ThreadLevel=1, ParentId=0, ChildId=0,  
   MsgTimeStamp, SenderId, RecipientName, Subject,  
   AckDate=0, Status=Unread,  
   Type=Thread, ExpiryDate, IsForward=No)

**FIG. 3A**  
 SUBSTITUTE SHEET (RULE 26)



4.2) Determine Recip't (ID) of Msg  
 4.3) Determine whether Recip't (ID)  
 is on system  
 4.4) IF ON, Then send MsgAlert to  
 Recip't

4.5) Display MsgAlert  
 4.6) Send Recip't response  
 to MsgAlert (OK, CANCEL)

4.7) WAIT for Recip't response  
 (NULL, OK, CANCEL)

4.8)  
 a) IF NULL, Then do not send  
 Msg and resend MsgAlert every 90  
 sec.  
 b) IF OK, Then send Msg and  
 Update Msgs Table  
 c) IF CANCEL, Then keep Msg  
 in abeyance on server, do not  
 resend MsgAlert and activate alert  
 light.

4.9) Display Updated PMB

FIG. 4A

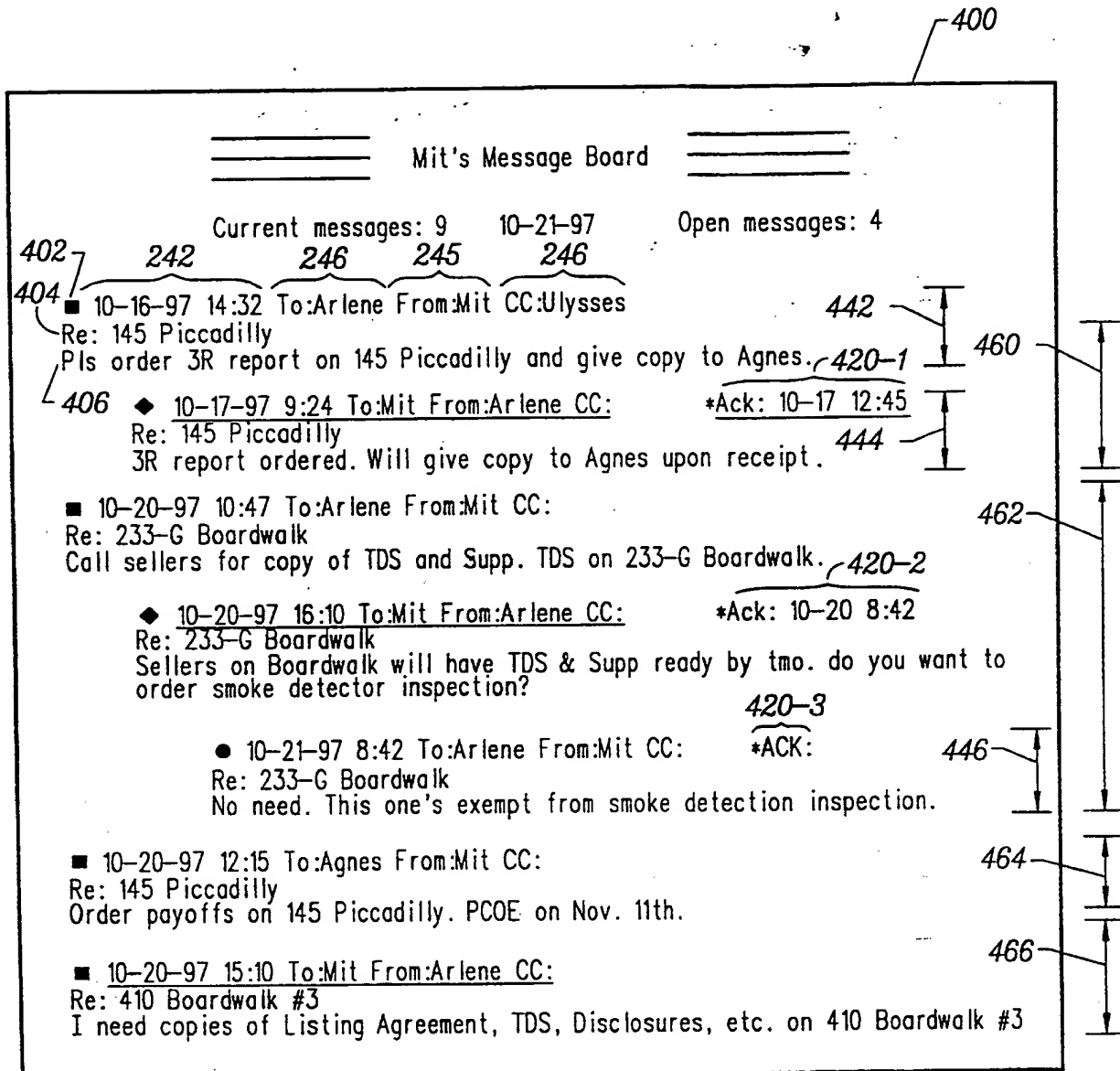


FIG. 4B



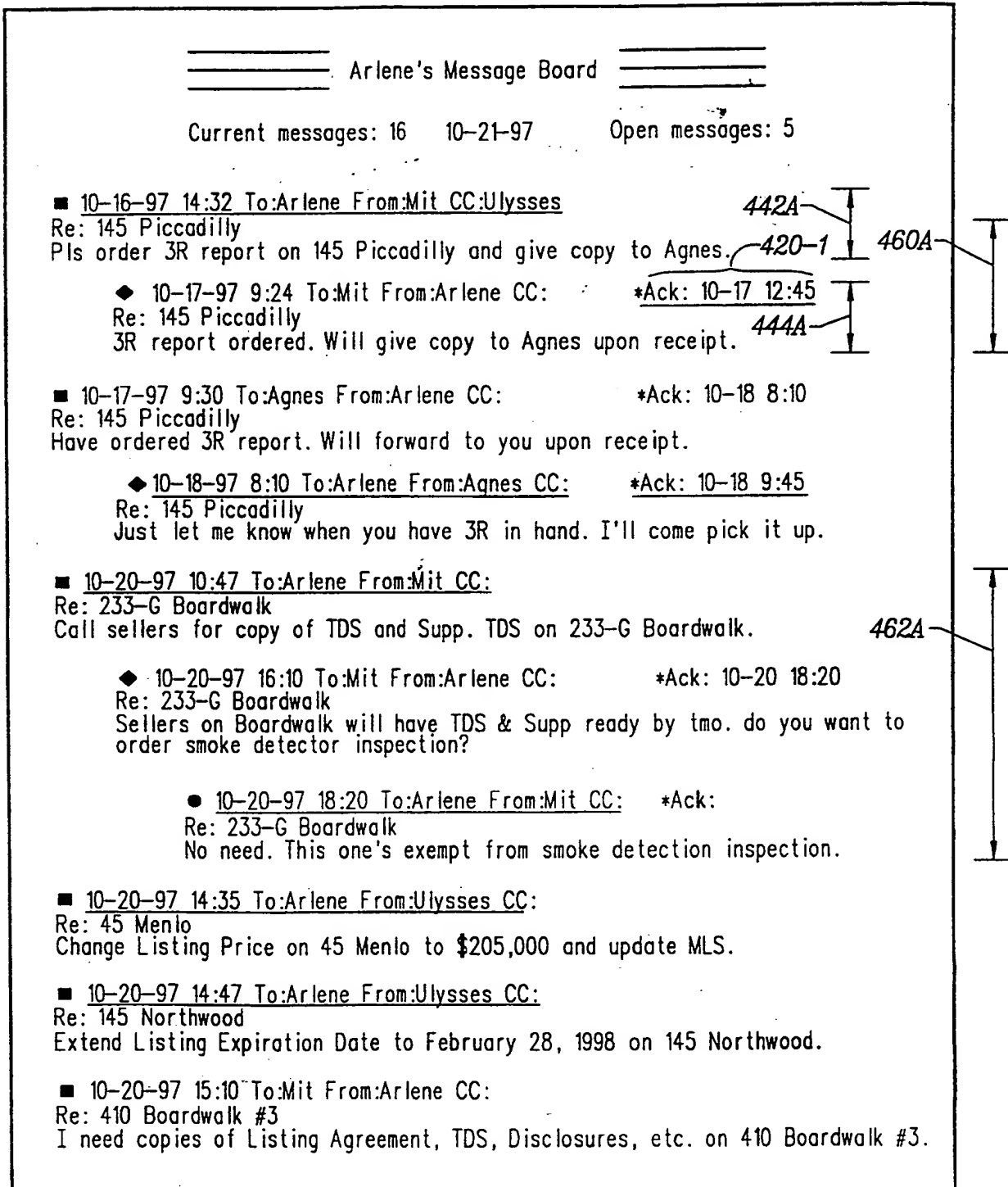


FIG. 4C

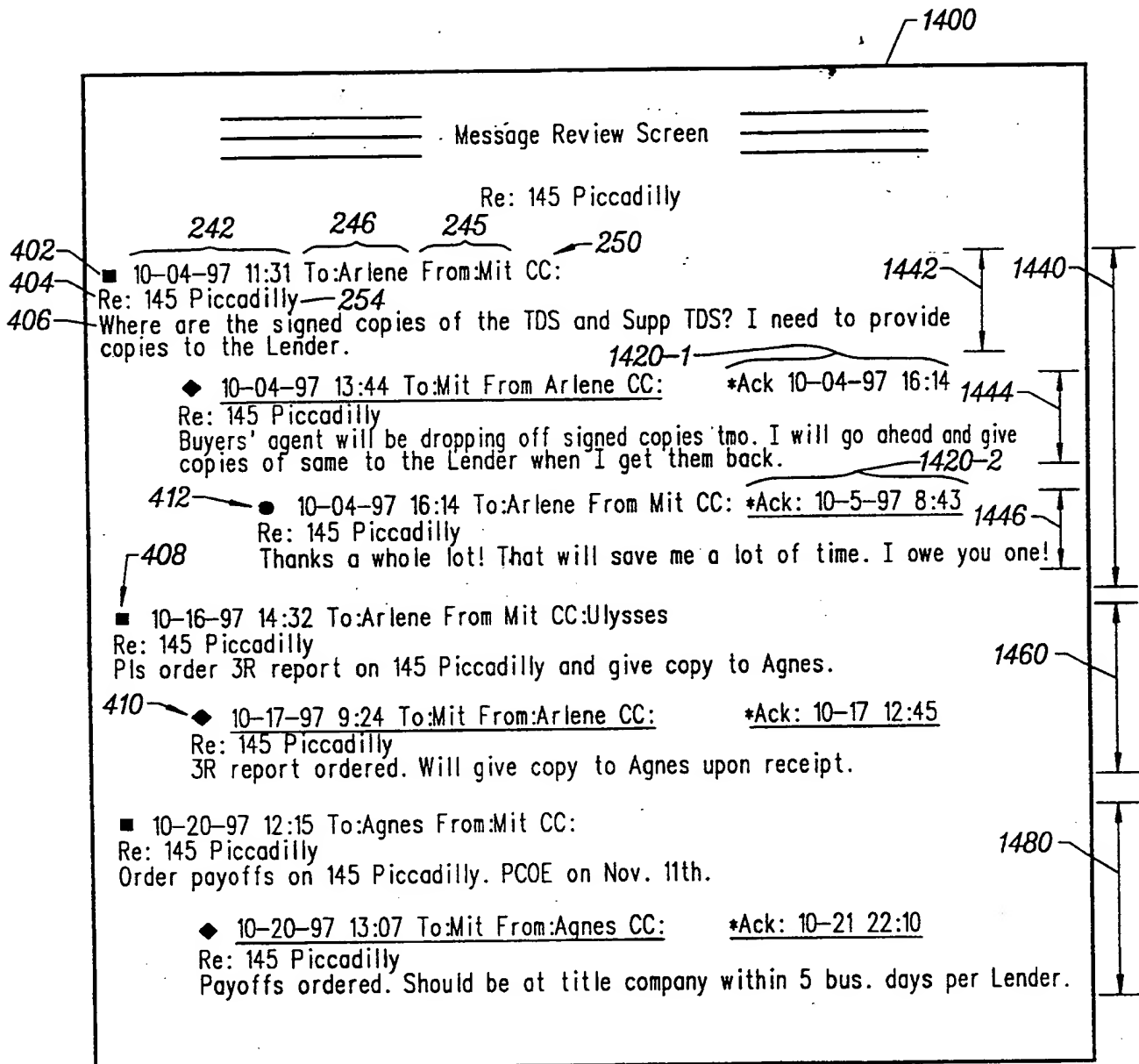
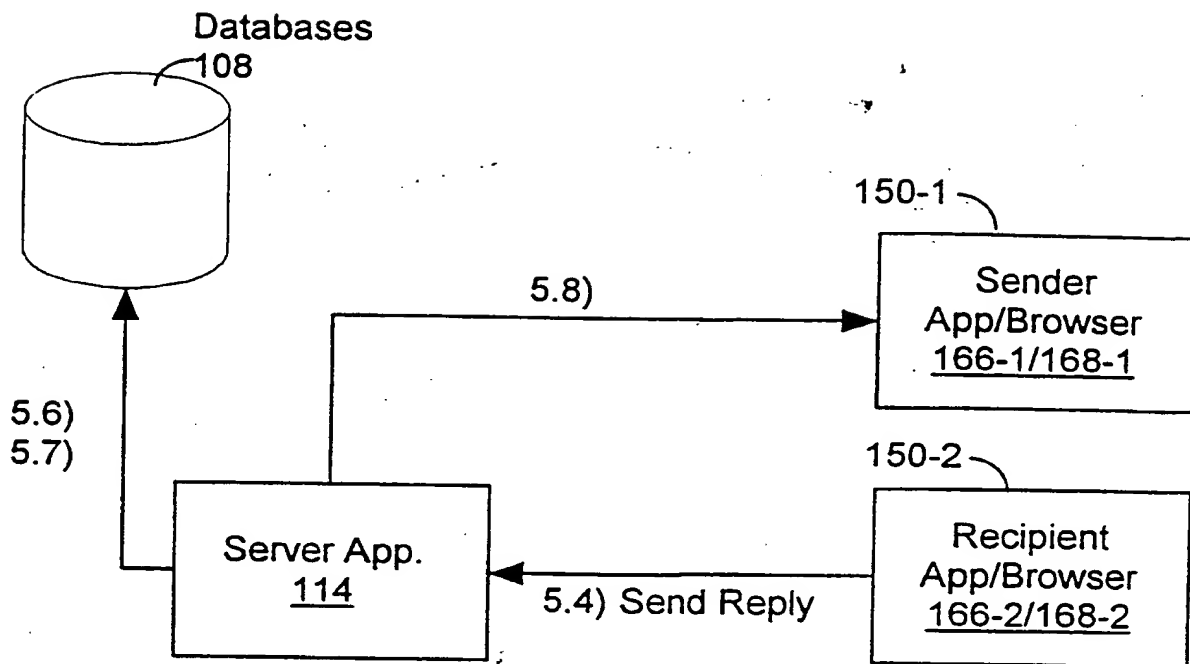


FIG. 4D



- 5.5) Assign unique ID to Reply
- 5.6) Store message in DB
- 5.7) Set parent/child pointers, update status in DB
- 5.8) Post Reply (and all other pending messages) to recipient IF recipient is online

- 5.1) Select Reply
- 5.2) Display Reply Box
- 5.3) User completes Reply
- 5.4) Send Reply and ID of message responded to

FIG. 5A

Threads Table 146

Thread Rec 1  
 ThreadID = T001  
 Subject = Lee Account  
 SubjectASCI = S050

Messages Table 142

Message Rec1 (M001)	<u>230-1</u>
Message Rec2 (M002)	<u>230-2</u>
ThreadLevel = 1	
ChildId = M005	
MsgRecOwner = ArnieS	
SenderName = ArnieS	
Recipient = SueG	
Message Rec3 (M003)	<u>230-3</u>
Message Rec4 (M004)	<u>230-4</u>
ThreadLevel = 1	
ChildId = M006	
MsgRecOwner = SueG	
SenderName = ArnieS	
Recipient = SueG	
Message Rec5	<u>230-5</u>
MsgId = M005	
MsgFN = M002.htm	
ThreadLevel = 2	
ThreadId = T001	
ParentId = M002	
MsgRecOwner = ArnieS	
SenderName = SueG	
Recipient = ArnieS	
Message Rec6	<u>230-6</u>
MsgId = M006	
MsgFN = M002.htm	
ThreadLevel = 2	
ThreadId = T001	
ParentId = M004	
MsgRecOwner = SueG	
SenderName = SueG	
Recipient = ArnieS	

Threads Participants Table 148

Thread Participant Rec 21  
 ParticipantId = P021  
 ThreadID = T001  
 UserID = U007

Thread Participant Rec 25  
 ParticipantId = P025  
 ThreadID = T001  
 UserID = U019

Thread Participant Rec 32  
 UserID = U027

Users Table 140

User Rec 7  
 UserId = U007  
 Name = ArnieS  
 DisplayName = Arnie  
 HoldAlerts = NO

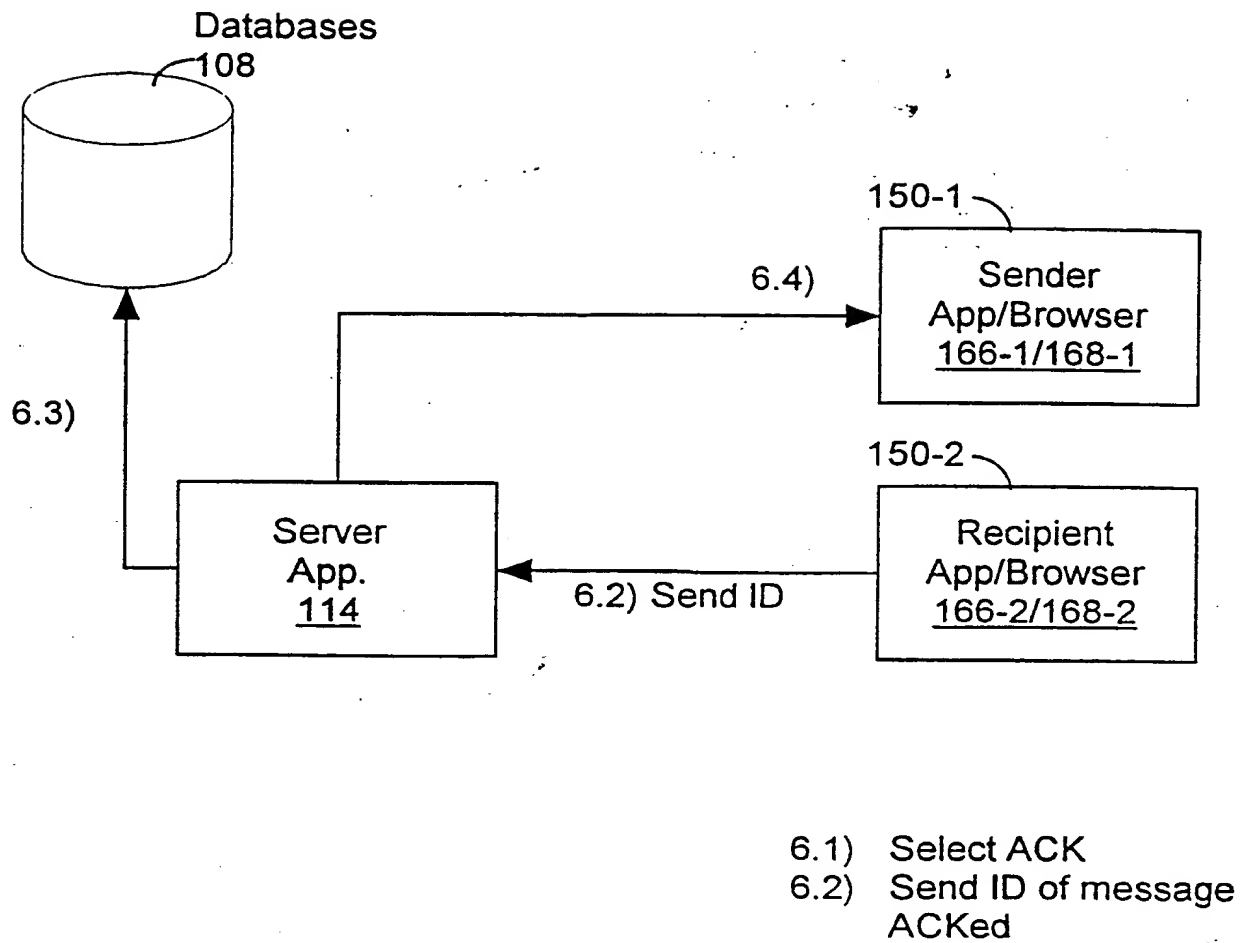
User Rec 19  
 UserId = U019  
 Name = BobB  
 DisplayName = Robert  
 HoldAlerts = YES

User Rec 27  
 UserId = U027  
 Name = SueG  
 DisplayName = SueEllen  
 HoldAlerts = NO

Hard Disk 104

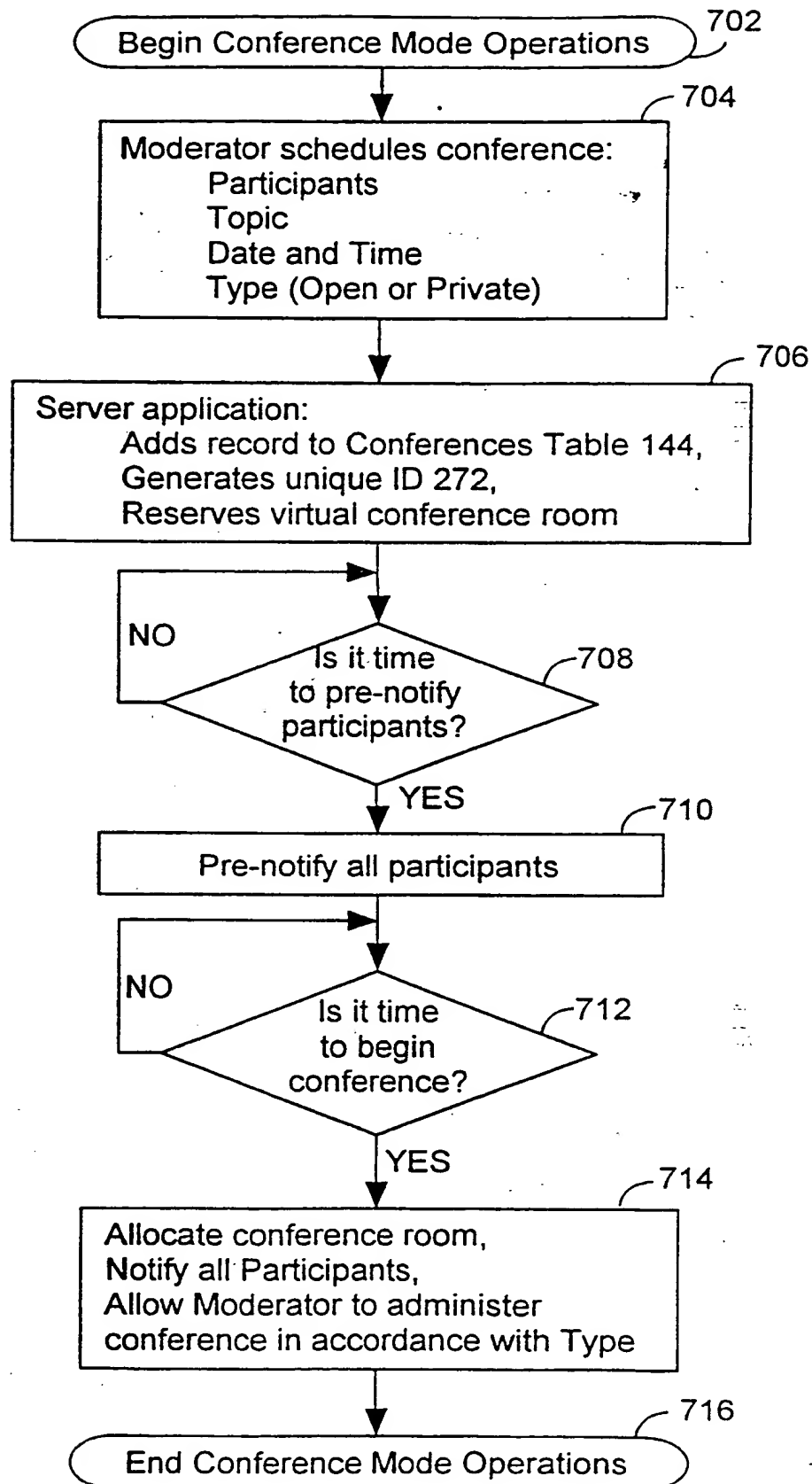
Msg Files	<u>136</u>
M001.htm: MsgTxt	
M002.htm: MsgTxt	

FIG. 5B



- 6.3) Update Msg status in DB (close thread)  
6.4) Post ACK to recipient IF recipient is online

FIG. 6



**FIG. 7A**  
SUBSTITUTE SHEET (RULE 26)

## CONFERENCE MODE IMPLEMENTATION

700

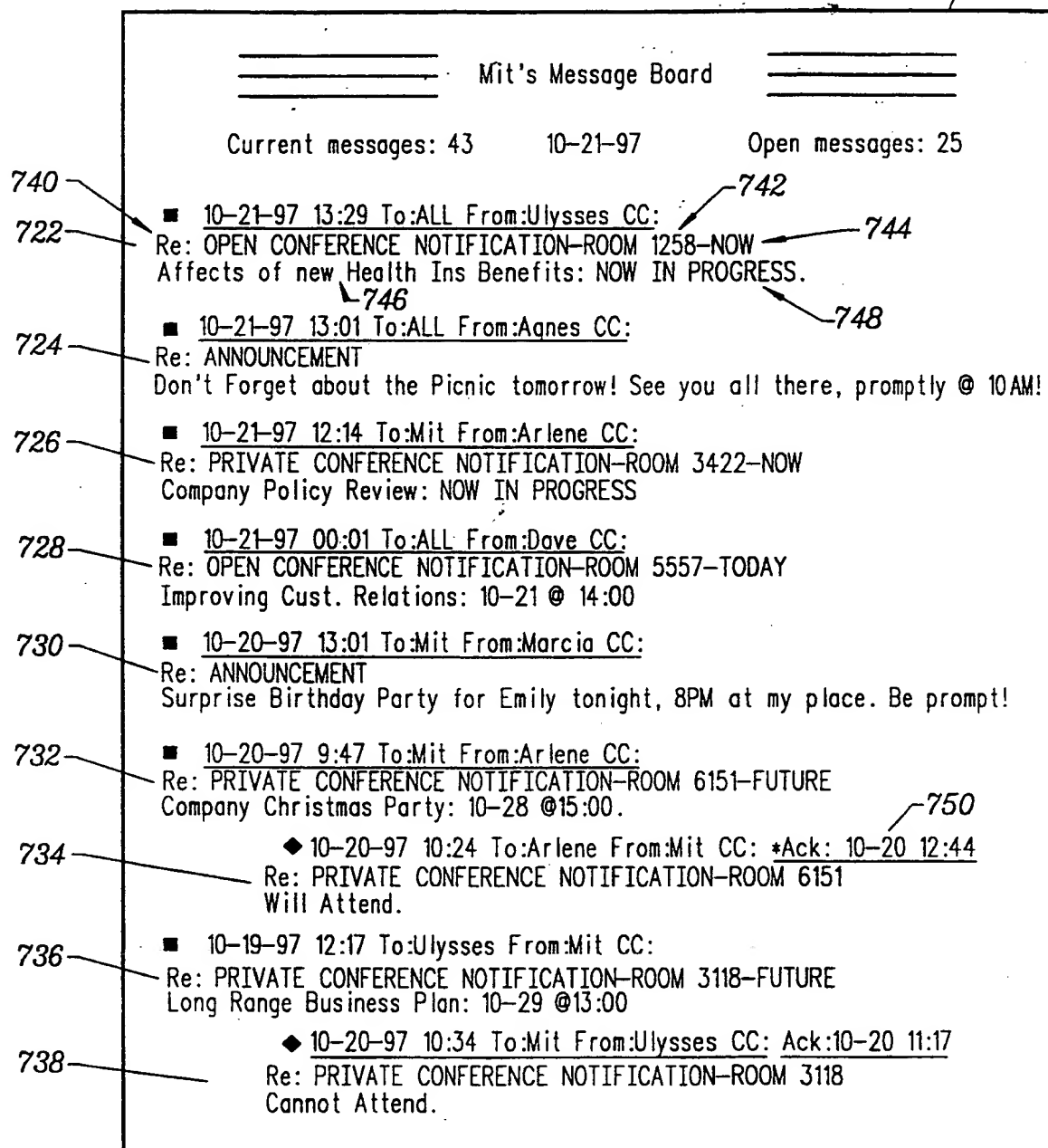


FIG. 7B

## CONFERENCE MODE IMPLEMENTATION

770

<div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>CONFERENCE SESSION</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>ROOM 6151</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>10-28-97</b> </div> <div style="width: 30%;"></div> </div>					
Display Board <span style="float: right;">772</span>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%; padding: 5px;"> <b>Topic: Company Christmas Party</b> </td> <td style="width: 40%; padding: 5px; text-align: right;"> <b>Moderator: Arlene</b> </td> </tr> <tr> <td colspan="2" style="padding: 10px;"> <div style="text-align: center;"> XYZ Corp.  Company Christmas Party  AGENDA </div> <div style="margin-top: 10px;"> I.     Logistics <div style="margin-left: 20px;"> A. Setup  B. Registration  C. Entertainment </div> </div> <div style="margin-top: 10px;"> II.    Finances <div style="margin-left: 20px;"> A. Band vs. D.J.  B. Dinner Costs  C. Santa Rental  D. Gift Exchange </div> </div> </td> </tr> </table>	<b>Topic: Company Christmas Party</b>	<b>Moderator: Arlene</b>	<div style="text-align: center;"> XYZ Corp.  Company Christmas Party  AGENDA </div> <div style="margin-top: 10px;"> I.     Logistics <div style="margin-left: 20px;"> A. Setup  B. Registration  C. Entertainment </div> </div> <div style="margin-top: 10px;"> II.    Finances <div style="margin-left: 20px;"> A. Band vs. D.J.  B. Dinner Costs  C. Santa Rental  D. Gift Exchange </div> </div>	
<b>Topic: Company Christmas Party</b>	<b>Moderator: Arlene</b>				
<div style="text-align: center;"> XYZ Corp.  Company Christmas Party  AGENDA </div> <div style="margin-top: 10px;"> I.     Logistics <div style="margin-left: 20px;"> A. Setup  B. Registration  C. Entertainment </div> </div> <div style="margin-top: 10px;"> II.    Finances <div style="margin-left: 20px;"> A. Band vs. D.J.  B. Dinner Costs  C. Santa Rental  D. Gift Exchange </div> </div>					
<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px 10px;">PRINT</div> <div style="border: 1px solid black; padding: 2px 10px;">HIDE</div> <div style="border: 1px solid black; padding: 2px 10px;">MODIFY</div> <div style="border: 1px solid black; padding: 2px 10px;">SEND</div> </div>					
Comment Screen <span style="float: right;">774</span>	<div style="display: flex;"> <div style="flex: 1; padding: 5px;"> <p>Frm:Arlene- I think we should ask Larry to play Santa again. He did such a good job last year. The kids loved him!</p> <p>Frm:Mit- Yes. I agree. Agnes, can you give him a call and ask him if he's available on Dec 19th?</p> <p>&gt;&gt;&gt;Ulysses has joined the session.&lt;&lt;&lt;</p> <p>Frm:Agnes- Okay. I'll call him tomorrow. What about caterers?</p> <p>Frm: Arlene- Nico did a wonderful job for our Appreciation Party last month: I think we should use him again!</p> <p>Frm:Ulysses- Hey, everyone, Sorry I'm late. Came from a long meeting.</p> <p>Frm:Mit- Glad you can join us: Uly! We need your input.</p> <p>&gt;&gt;&gt;Sammy has left the session.&lt;&lt;&lt;</p> <p>Frm:Arlene- Uly, I posted an agenda on the board for everyone to see. Wer're on #2 right now. Well, kinda.</p> </div> <div style="flex: 0.2; border-left: 1px solid black; border-right: 1px solid black; padding: 5px; text-align: center;"> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div> <div style="width: 40%; text-align: center;"> <b>Participants</b> </div> <div style="width: 30%;"></div> </div> <div style="display: flex; justify-content: space-between; border-bottom: 1px solid black; margin-bottom: 5px;"> <div style="width: 30%;"></div></div></div></div>				



## TALK MODE IMPLEMENTATION

800

TALK SESSION  
Dialogue Input Screen

To:

802

804

Hey, Mit. Did you hear the rumor about the Christmas Party?

806

FIG. 8A

## TALK MODE IMPLEMENTATION

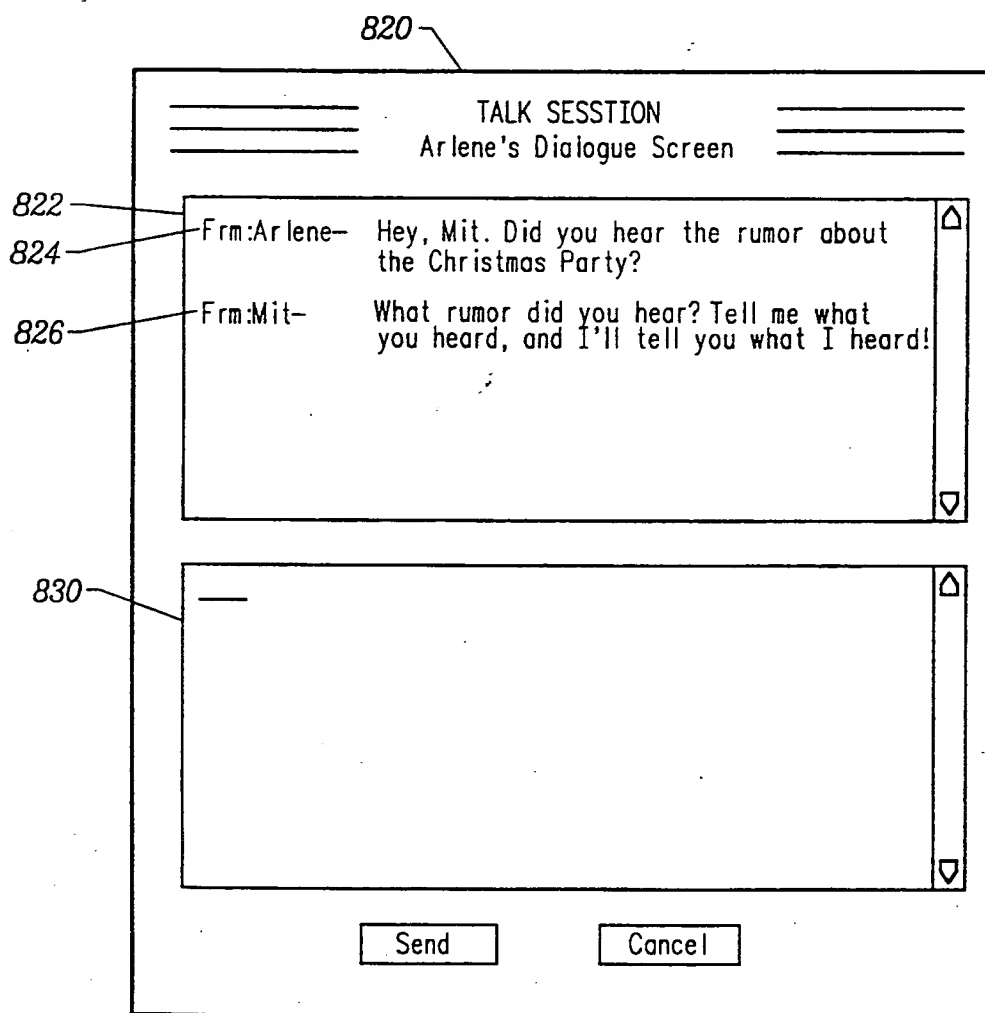


FIG. 8B

## MAIL MODE IMPLEMENTATION

		Mit's Mail Folder			
Folder: Personal/misc		10-27-97	eMail Account: mitmaur		
Mail You've Received: 47					
902	Date/Time	Subject	912	Sender	Size
906	<input checked="" type="checkbox"/> 10-27 1:16Pz	<u>Toastmaster's Meeting</u>		DVaughn@d4tm.org	23K <input type="checkbox"/> <input checked="" type="checkbox"/>
	906a	<input type="checkbox"/> 10-29 12:55P 2K			
908	<input checked="" type="checkbox"/> 10-27 11:43A	<u>Looking for property to buy</u>		tomthumb@aol.com	2K <input type="checkbox"/> <input checked="" type="checkbox"/>
	908a	<input type="checkbox"/> 10-29 1:16P 6K			
910	<input checked="" type="checkbox"/> 10-27 8:15A	<u>New pricing schedule</u>		HYoung@onol.com	18K <input type="checkbox"/> <input checked="" type="checkbox"/>
	<input checked="" type="checkbox"/> 10-26 5:36P	<u>Company Hiring Policy Review</u>		amy_wang@polka.com	34K <input type="checkbox"/> <input checked="" type="checkbox"/>
	910a	<input type="checkbox"/> 10-26 8:37A 4K			
	<input checked="" type="checkbox"/> 10-26 3:27P	<u>Revise Monthly Statement</u>		WBll@cnet.com	18K <input type="checkbox"/> <input checked="" type="checkbox"/>
	<input checked="" type="checkbox"/> 10-26 3:05P	<u>1998 Company Picnic</u>		HLoc@webtv.com	22K <input type="checkbox"/> <input checked="" type="checkbox"/>
Mail You've Sent: 22					
904	Date/Time	Subject	To	Size	
	<input checked="" type="checkbox"/> 10-27 1:16P	<u>News about Maggie</u>	Ross@Bayinsider.com	6K	<input type="checkbox"/> <input checked="" type="checkbox"/>
	<input type="checkbox"/> 10-27 12:31P	<u>Calendar Orders will be delayed...</u>	Benas@work4u.com	11K	<input type="checkbox"/> <input checked="" type="checkbox"/>
	<input type="checkbox"/> 10-26 11:19A	<u>Mom's Cooking</u>	minortom@aol.com	3K	<input type="checkbox"/> <input checked="" type="checkbox"/>
	<input checked="" type="checkbox"/> 10-25 3:09P	<u>Cool Web Sites I Found</u>	BillyGatos@msn.com	38K	<input type="checkbox"/> <input checked="" type="checkbox"/>
	<input checked="" type="checkbox"/> 10-21 12:06P	<u>Vacation Plans</u>	MNubla@kron.com	12K	<input type="checkbox"/> <input checked="" type="checkbox"/>
	<input checked="" type="checkbox"/> 10-16 7:23A	<u>Operating Committee Meeting</u>	DVaughn@d4tm.org	22K	<input type="checkbox"/> <input checked="" type="checkbox"/>
	<input type="checkbox"/> 10-16 8:12A	<u>Reader's Digest article about the...</u>	HJHughes@ccnet.com	5K	<input type="checkbox"/> <input checked="" type="checkbox"/>
	<input type="checkbox"/> 10-12 1:09P	<u>Marketing Presentations</u>	CFigueroa@webtv.com	29K	<input type="checkbox"/> <input checked="" type="checkbox"/>
Move Groups		Sort Folder	Edit Folder	Cancel	
Check Other Folder		Check Other Account			

FIG. 9

## INTERNATIONAL SEARCH REPORT

 International application No.  
 PCT/US99/06074

## A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : G06F 13/00

US CL : 709/206

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 709/206

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS

search terms: chat line, email, conferencing, AOL, bulletin boards

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,206,934 A (NAEF, III) 27 April 1993, see entire document.	1-54
A	US 5,208,912 A (NAKAYAMA et al) 04 May 1993, see entire document.	1-54
A	US 5,404,488 A (KERRIGAN et al) 04 April 1995, see entire document.	1-54
A	US 5,577,247 A (UYAMA et al) 19 November 1996, see entire document.	1-54
A	US 5,600,834 A (HOWARD) 04 February 1997, see entire document.	1-54
A	US 5,630,116 A (TAKAYA et al) 13 May 1997, see entire document.	1-54

☒ Further documents are listed in the continuation of Box C.
 ☐ See patent family annex.

* Special categories of cited documents.	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
*A* document defining the general state of the art which is not considered to be of particular relevance	*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
*E* earlier document published on or after the international filing date	*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
*L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*Z* document member of the same patent family
*O* document referring to an oral disclosure, use, exhibition or other means	
*P* document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

20 MAY 1999

Date of mailing of the international search report

03 JUN 1999

 Name and mailing address of the ISA/US  
 Commissioner of Patents and Trademarks  
 Box PCT  
 Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

JASON D. CARDONE

Telephone No. (703) 305-3800

## INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US99/06074

## C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,684,984 A (JONES et al) 04 November 1997, see entire document.	1-54
Y,P	US 5,799,151 A (HOFFER) 25 August 1998, column 11, line 50 - column 12, line 37 and column 16, line 45 - column 18, line 19.	1-54
Y,P	US 5,862,325 A (REED et al) 19 January 1999, columns 8-10, column 13, line 7 - column 14, line 55, column 18, lines 27-51, column 22 - 23, and column 144 -145	1-54

**This Page Blank (uspto)**